COMPUTER AIDED DESIGN OF SOME MOS LSI / VLSI FUNCTIONAL MODULES

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

By
T. H. SREENIVAS

to the
DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

JANUARY, 1986

CERTIFICATE

Certified that the work entitled 'COMPUTER AIDED DESIGN OF SOME MOS LSI/VLSI FUNCTIONAL MODULES' by Mr. Sreenivas, T.H. under our supervision has not been submitted elsewhere for a degree.

Dhande

Dr. S.G. Dhande
Professor
Dept. of Mechanical
and
Computer Science Engg.
IIT Kanpur - 16

M Hasan.

Dr. M.M. Hasan
Professor
Dept. of Electrical Engg.
IIT Kanpur - 16

ACKNOWLEDGEMENTS

I take this opportunity to thank Dr. M.M. Hasan and Dr. S.G. Dhande for their inspiring guidance and support throughout this work.

My heartfelt thanks are due to Dr. V.P. Sinha, for the valuable discussions I had with him.

I would like to express my gratitude to Mr. U.A. Vinay Kumar and Mr. K.G. Shastry - research scholars of the institute, for correcting the manuscript and proofs.

Finally, I would like to thank Mr. Yogendra, for the neat typing of the thesis.

IIT, Kanpur

December, 1985.

Sreenivas, T.H.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Three software packages, named 'Computer-Aids for Random Logic Implementation', 'Programmable logic Array (PLA) stick diagram generator' and 'PLA layout generator' have been developed using Fortran, Pascal programming languages and PLOT-10 Interactive Graphics Library (IGL) routines.

Computer-aids include detailed descriptions of standard or frequently used gates (Inverter, NOR and NAND gates, etc.) stored on the device library. More complex structures such as XOR gate, flip-flop, shift-register etc. are easily constructed of the basic building blocks stored on the device library and metal interconnections.

The PLA stick diagram generator generates the stick diagram (or symbolic layout) of the desired logic function, from the multi-input sum of products expression of the logic function.

The PLA layout generator draws the layout of the logic function from its sum of products expression.

The purpose of this work is to help the IC chip designer in reducing the design time, lowering the cost of design, eliminating the layout errors and in making a transition from the logic diagram or circuit schematic to the camera ready artwork.

The packages have been tested on some functional modules like the Full Adder, the Decade Counter, the Binary to Gray Code Converter, the Multiplexer, etc., and the same have been simulated for their functional behaviour and electrical characteristics using SPICE program.

CHAPTER 1

INTRODUCTION

1.1 Evolution of Integrated Circuits

The ERA of integrated circuits began in the late 1950's with the discovery and development of the planar silicon technology that opened the way for the integration of an ever-increasing number of circuit functions. The evolution of integrated circuits has been characterized by small-scale integration (SSI), medium-scale integration (MSI), and large scale integration (LSI). In the Future, it will be possible to place a million or more circuit elements on a single chip at a cost hardly higher than that of a single vacuum tube.

SSI, in the early 1960's, was characterized by an integration complexity of several logic gates per chip. The MSI phase began in the second half of the sixties with the development of a stable MOS process. The highest complexity MSI chip corresponded to about 100 logic gates or 256 bits of memory. The beginning of LSI phase is generally identified with the introduction of the 1K-bit dynamic MOS RAM, in the year 1970. LSI can be characterized by the development of proved silicon process technology.

The transition to the VLSI phase is generally associated with the development of 64K-bit dynamic MOS RAM'S, 16K-bit static MOS RAM'S, 16-bit microprocessors, and logic arrays with

about 10,000 logic gates.

As predicted by G.E. Moore (Moore's Law), the densities of semiconductor memories have been doubling every year due to more clever designs, more complex structures, and smaller structural dimensions. The VLSI phase, must therefore be characterized by the development of process technologies, device structures, and the circuit and system design techniques to continue the drastic reduction of structural dimensions that will be necessary to fullfil the law of doubling the density each year.

With the push towards submicron technology, transistor models have become increasingly complex. The number of components in integrated circuits has forced designer's efforts and skills towards higher levels of design. The ability to place larger systems on a single piece of silicon has made more advanced systems realizable. However, this increase in capability brings with it new problems for the designer.

One major problem is how to manage the design and analysis of systems with a large number of components. Potential solutions to this problems include developing methodologies which exploit regularity and hierarchy during the design phase, and computer aided design tools which take over some of the lower-level tasks of design and analysis. Much of the design takes place at an abstract level, for example the MOSFET is

modelled as a charge controlled switch. This allows the designer to ignore unnecessary details.

1.2 Computer Aided Design (CAD)

Since artwork is possible at the basic level of design only, viz., polygon , the topology of the LSI chip is manually created by drawing shapes, polygon-by-polygon, on layers of Mylar. The layout is then digitized point by point and entered into the memory. The power of the computer is thereupon not helpful to perform tasks such as design-role or interconnection.

Most of the LSI/VLSI manufacturers substitute these methods by artwork techniques totally based on CRT-display, and software for individual stages of design.

For example, the logic cells stored in computer memory are retrieved to construct a larger array of logic. The entire chip composition is methodically automated stage-by-stage to complete the design of the high-density microprocessor or memories.

Hewlett-Packard Co. uses interactive computer graphics with light-pen to draw the schematic level of the design on the screen with the aid of multi-level polygon representation and then continue the rest of the design with software.

In Japanese CAD, software programs that generate layouts replace manual drawing and digitizing. Compared to America, Japan uses more powerful computers to run CAD design, simulation, and debugging of software. Nippon Electric Company has a program to perform logic-checking from the layout by comparing circuits with drawn simulated circuitry.

1.3 Design Techniques

Compared with random logic circuits, memory-type circuits are more suitable for LSI realization since their iterated structure of identical cells results in higher transistor density and higher yield. A programmable logic array (PLA) is a read only memory (ROM) with programmable addresses and it is suitable for realizing logic functions with many unspecified input connections.

In 1947 a study was initiated to evaluate the possibilities of a set of PLA based macros in implementing small processors. In that study an 8-bit microprocessor was redesigned using a set of blocks including PLA's, registers (of several types) and busses. The study took the position that the merits of PLA - based design were to be measured by the number of bits equivalent to a NOR circuit. The result for the particular microprocessor was a PLA design that offered equivalent performance and density at one-third the power of the original Weinberger NOR design. Since the PLA cost is mainly determined

by the number of pins and chip area,both of which are affected by the number of inputs, the reduction of the number of inputs is very important in PLA design. On the other hand, the reduction of the number of product terms in a sum-of-product expression is important in conventional random logic synthesis.

Chapter 2 introduces the basics of MOSFETS and the primitive cells of NMOS logic. Structured and unstructured design approaches are discussed with emphasis on the PLA.

Chapter 3 describes the implementation of the three packages, viz., 'Computer Aids for Random Logic Implementation', 'Stick Diagram Generator', and PLA Layout Generator'.

Chapter 4 deals with the simulation of the circuits using the circuit simulator-SPICE.

Chapter 5 has computer generated stick diagrams and layouts of several functional cells, using the packages that have been developed.

Appendix - A is on the User's Guide. Here an elaborate explanation of the facilities available on the packages and instructions on using the packages are given.

Appendix - B is about the Graphics package - PLOT-10, Interactive Graphics Library (IGL).

Appendix - C is about the usage of the SPICE package.

CHAPTER 2

MOS INTEGRATED CIRCUITS

2.1 Introduction to MOS Technology

Integrated systems in metal-oxide-semiconductor (MOS)-technology contain three levels of conducting material separated by intervening layers of insulating material. Going from top to bottom, the levels are called metal, polysilicon, and diffusion respectively. Contact cuts are made in the insulating material to connect certain points between levels.

In the absence of contact cuts through the insulating material, paths on the metal level may cross over paths on either polysilicon level or diffusion level with no significant functional effect. However, whereever a path on the polysilicon level crosses a path on the diffusion level, a transistor is created. Such a transistor has the characteristic of a simple switch, with a voltage on the polysilicon-level controlling the flow of current in the diffusion level path. Circuits composed of such transistors, interconnected by paths on the three levels, form the basic building blocks. With these basic circuits, integrated circuits to be fabricated on the surface of monolithic crystalline chips of silicon are designed.

2.2 The MOS Transistor

The circuit schematic of a MOS transistor is shown in Fig. 2.1. An MOS transistor is produced on the integrated

system chip, whenever a polysilicon path crosses a diffusion

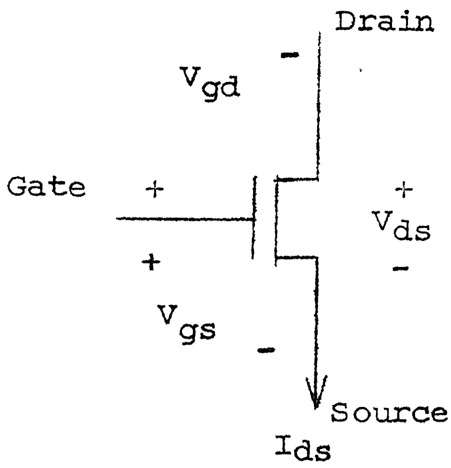


Fig. 2.1 MOS transistor symbol

path as shown in fig. 2.2. For the n-channel metal-oxide semiconductor field-effect transistors (MOSFET), the terminal labels are assigned such that the drain-to-source voltage V_{ds} is normally positive. During the fabrication, diffusion paths are formed

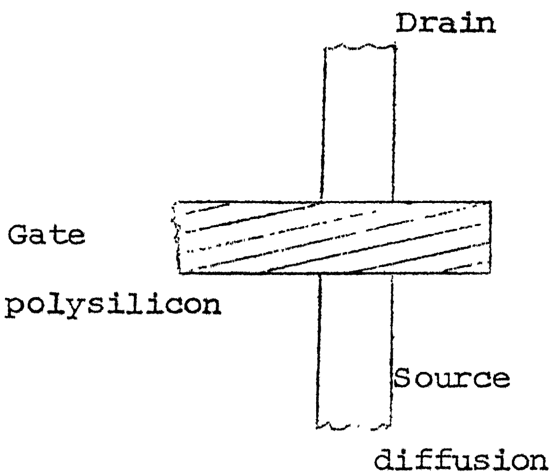


Fig. 2.2 MOS transistor - top view

after the polysilicon paths.

The poly gate and the thin layer of oxide beneath it, mask the region under the gate during diffusion. Therefore, no diffusion path forms under the gate and there is no direct connection on the diffusion level between the source and the drain terminals of the transistors.

In the absence of any charge on the gate, the drain-to-source path through the transistor is like an open switch. The gate separated from the substrate by the layer of thin oxide, forms a capacitor. If sufficient charge is placed on the gate so that gate-to-source voltage V_{gs} exceeds a threshold voltage V_{th}, electrons are attracted to the region under the gate to form a thin conducting path between the drain and the source. Most of the transistors have threshold voltages greater than zero. Such transistors are called enhancement mode MOSFETS and their threshold voltage is approximately 0.2 VDD, where VDD is the positive supply voltage for the particular technology.

The relationship between drain-to-source current I_{ds}, drain-to-source voltage V_{ds}, and gate-to-source voltage V_{gs} is sketched in Fig. 2.3.

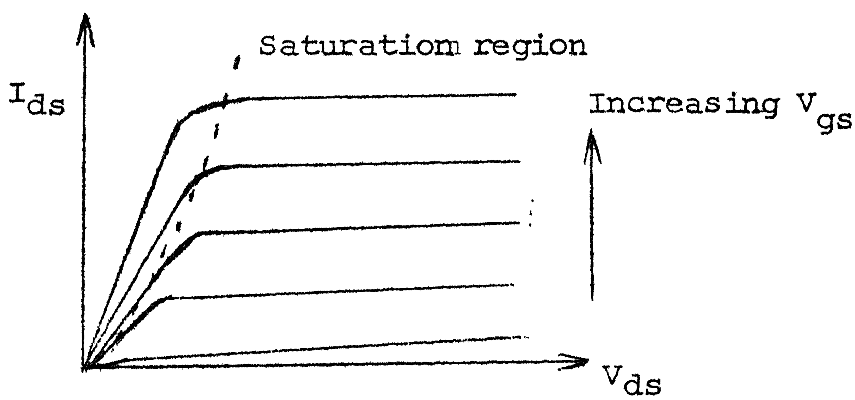


Fig. 2.3 Current Vs. Voltage characteristics of a MOSFET

It can be seen from Fig. 2.3, that for small V_{ds}, the drain current is proportional to the source-drain voltage and also to the gate voltage above the threshold. Any device with a current through it proportional to the voltage across it may be viewed as a resistor, and in the case of an MOS device with low drain-to-source voltage, the resistance is controlled by the gate voltage.

For $V_{ds} > (V_{gs} - V_{th})$, the drain current becomes independent of V_{ds}. A further increase in V_{ds} does not increase I_{ds}. This range of V_{ds} is known as saturation region.

The drain-source current I_{ds} and the drain-source voltage V_{ds} relationship is given by,

$$I_{ds} = \beta \ [(V_{ds} - V_{th}) \ V_{ds} - \frac{1}{2} \cdot V_{ds}^2] \quad \text{in linear region}$$

$$I_{ds} = \beta/2 [\ V_{gs} - V_{th}\ ]^2 \quad \text{in saturation region}$$

where,

β = transistor gain factor = K.W/L

K = conduction factor.

$$= \mu C_o = \frac{\mu \ \varepsilon_{ox} \ \varepsilon_o}{t_{ox}}$$

W/L = Aspect ratio of MOSFET

W = Width of channel

L = length of the channel

μ = mobility of the charge carrier, electrons in case of N-channel MOSFET and holes in case of P-channel MOSFET

t_{ox} = oxide thickness.

2.3 The Basic Inverter

The inverter produces an output which is the complement of the input. While describing the logic function of circuits in integrated systems, the value logic-1 is assigned to voltages equaling or exceeding some defined logic threshold voltage, and logic-0 is assigned to voltages less than the threshold voltage.

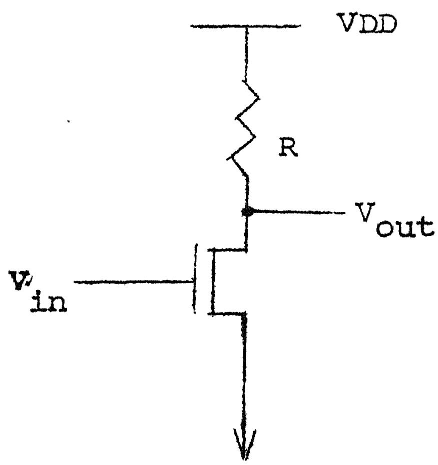
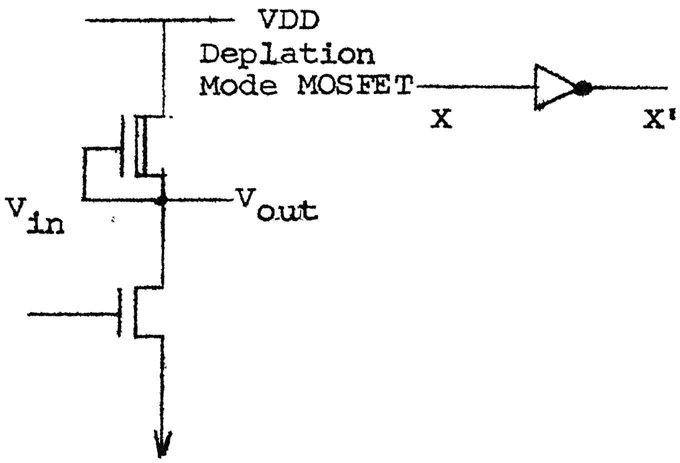


Fig. 2.4 The basic inverter

Had there been an efficient way to implement resistors im MOS technology, an inverter could have been built using the configuration of Fig. 2.4. For the input voltage V_{in} less than the transistor threshold voltage V_{th}, the transistor is switched off and V_{out} is 'pulled up' to the positive supply voltage VDD. If V_{in} is greater than V_{th}, the transistor is switched off and current flows from VDD supply through the resistor R to the ground. If R were sufficiently large, V_{out} could be 'pulled down'

well below V_{th}, complementing the input. But, the resistance per unit length of minimum-width lines of various available conducting elements is far less than the effective resistance of the switched-on MOSFET. Implementing a very large pull-up using resistive lines would require a very large area compared to that occupied by the transistor itself.

To overcome this problem, a depletion mode transistor is used as a pull-up for the basic inverter circuit symbolized and configured as shown in Fig. 2.5. The depletion mode MOSFET,



X	X'
1	0
0	1

Fig. 2.5 The inverter circuit diagram, logic symbol, and truth table.

in contrast to the enhancement mode transistor, has a threshold voltage V_{dep} less than zero. The voltage on the gate of a depletion mode transistor relative to its source must be less than V_{dep} for it to turn off. But the gate is always connected to

source and the transistor remains on always.

The computer generated layout of the inverter is shown in Fig. 2.6. Two polysilicon regions crossing a path in the diffusion level running between VDD and ground are seen in the layout. This arrangement forms the two MOS transistors of the inverter. The inverter input A is connected to the poly that forms the gate of the lower transistor. The pull-up is formed by shorting the gate of the upper transistor to its source (design rules are described in Chapter 3). The output of the inverter is taken from the diffusion level between the drain of the pull-down and the source of the depletion mode pull-up transistor.

2.4 The NOR and the NAND Gates

In Fig. 2.7(a) and 2.7(b), the circuit schematic of a 2-input NOR gate with $\beta_R = 8:1$ (β_R = (W/L) driver/(W/L) load). In order to maintain appropriate logic levels(particularly the low logic output level), the NAND configuration requires pull-down transistors with twice the aspect ratio, each corresponding to one half of the resistance of the pull-down transistor of the inverter. The fall time of the NAND structure approximates to that of the inverter. The fall time of the 2-input NOR gate for the condition when all the inputs are high, will be twice as fast as the inverter. In view of the increased input

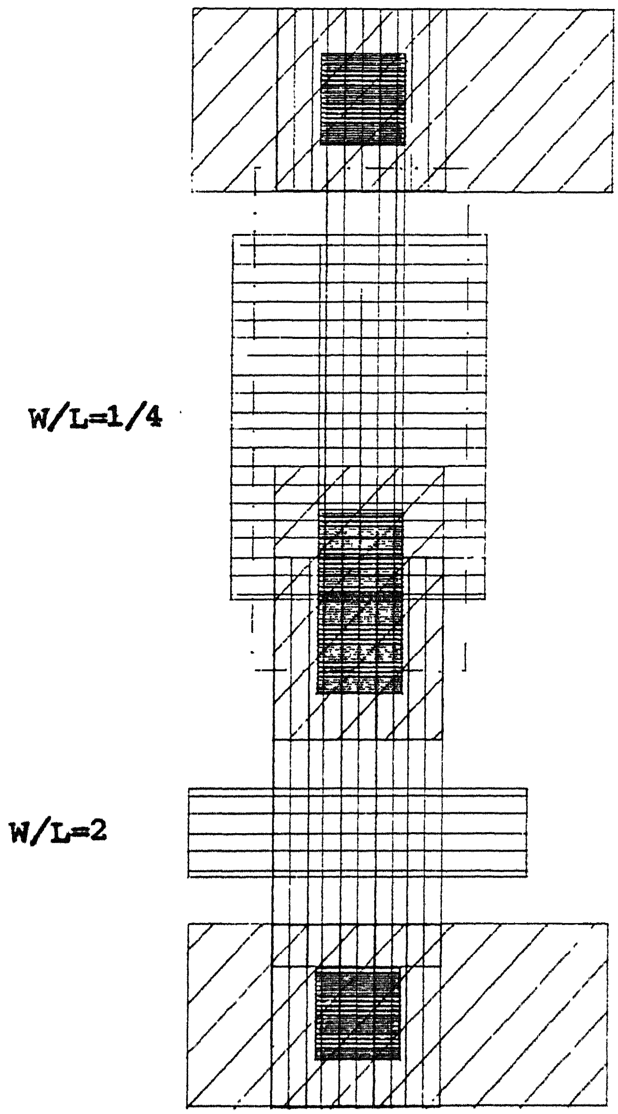


Fig.2.6 Layout of an Inverter

gate areas and thus higher input capacitance of the NAND structure, the NAND arrangement is not favoured and logic minimization techniques based on NOR logic are encouraged in MOS large scale integration (LSI).

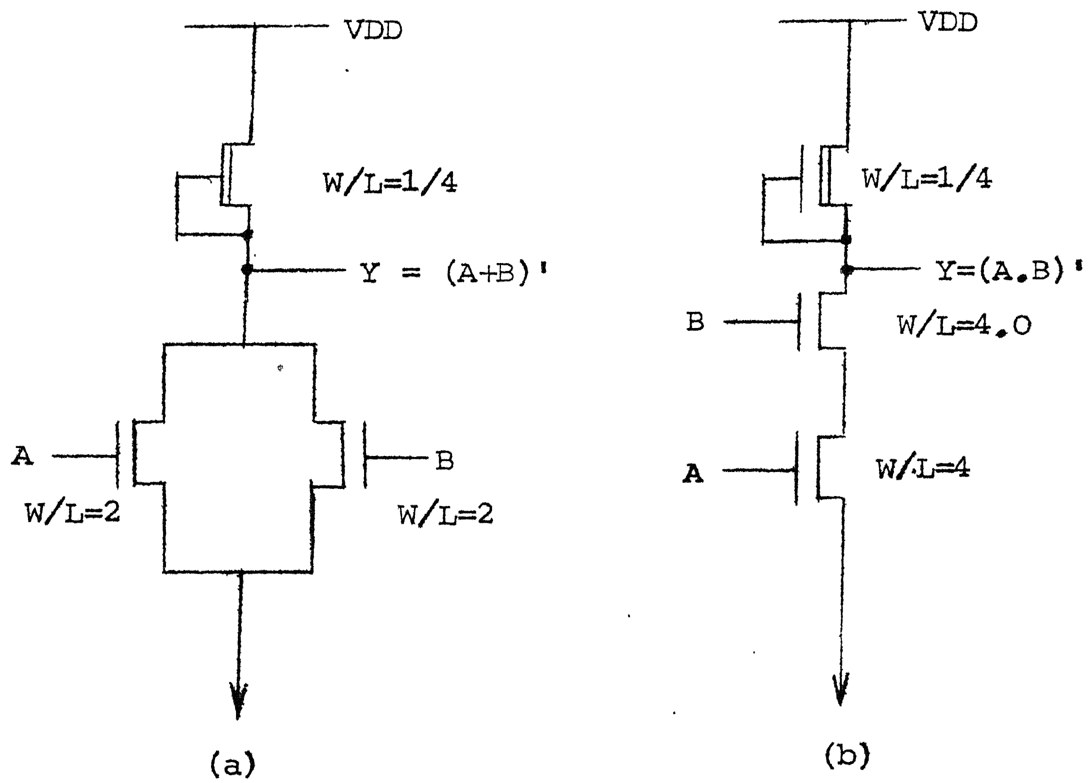


Fig. 2.7(a) 2-input NOR gate (b) 2-input NAND gate

The layouts of a two-input NOR gate and a two-input NAND gate generated on the computer are shown in figures 2.8 and 2.9 respectively.

2.5 Logic Design With MOS

Four main circuit design methods are available in MOS LSI. They are,

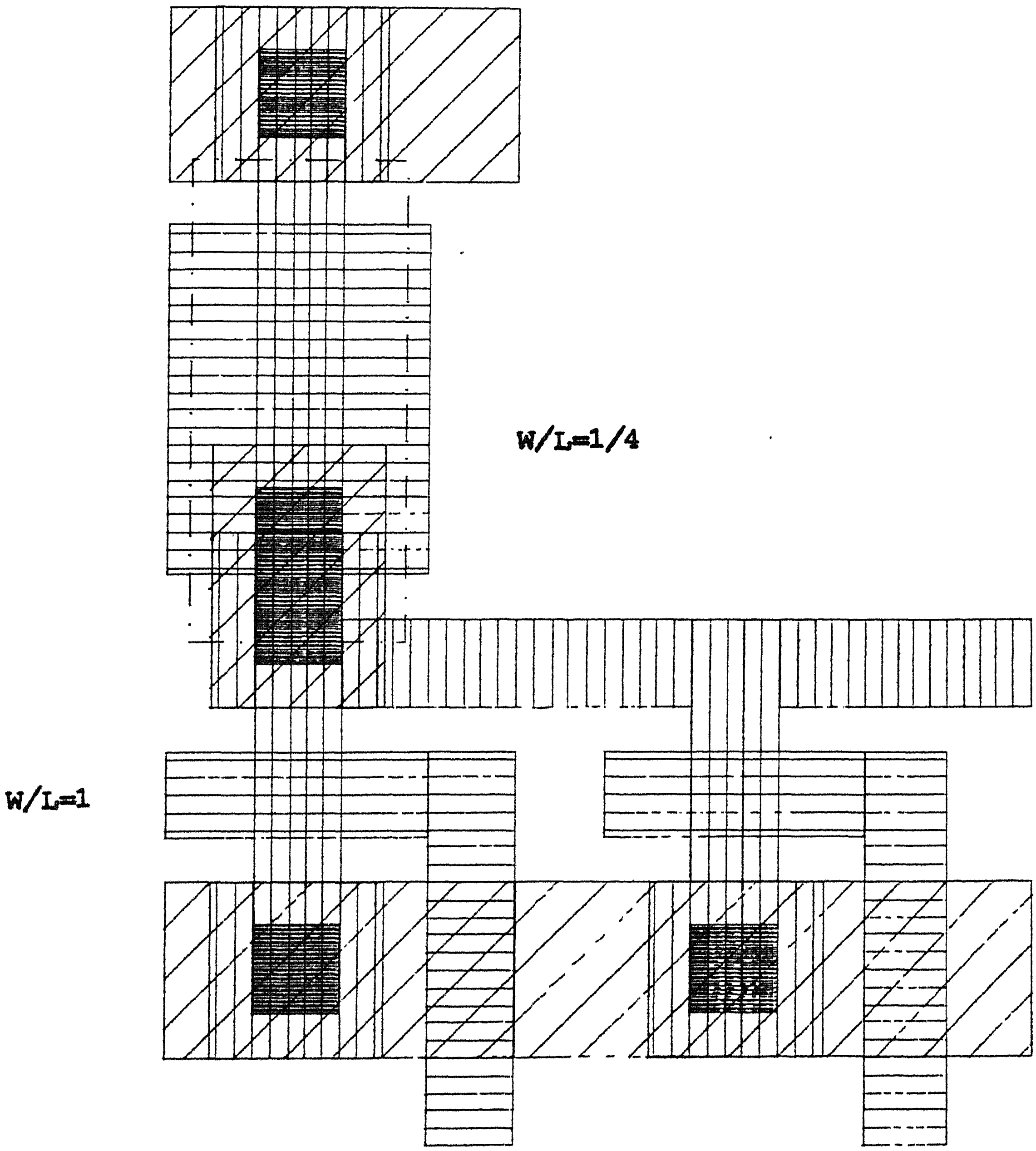


Fig. 2.8 Layout of a 2-input NOR Gate

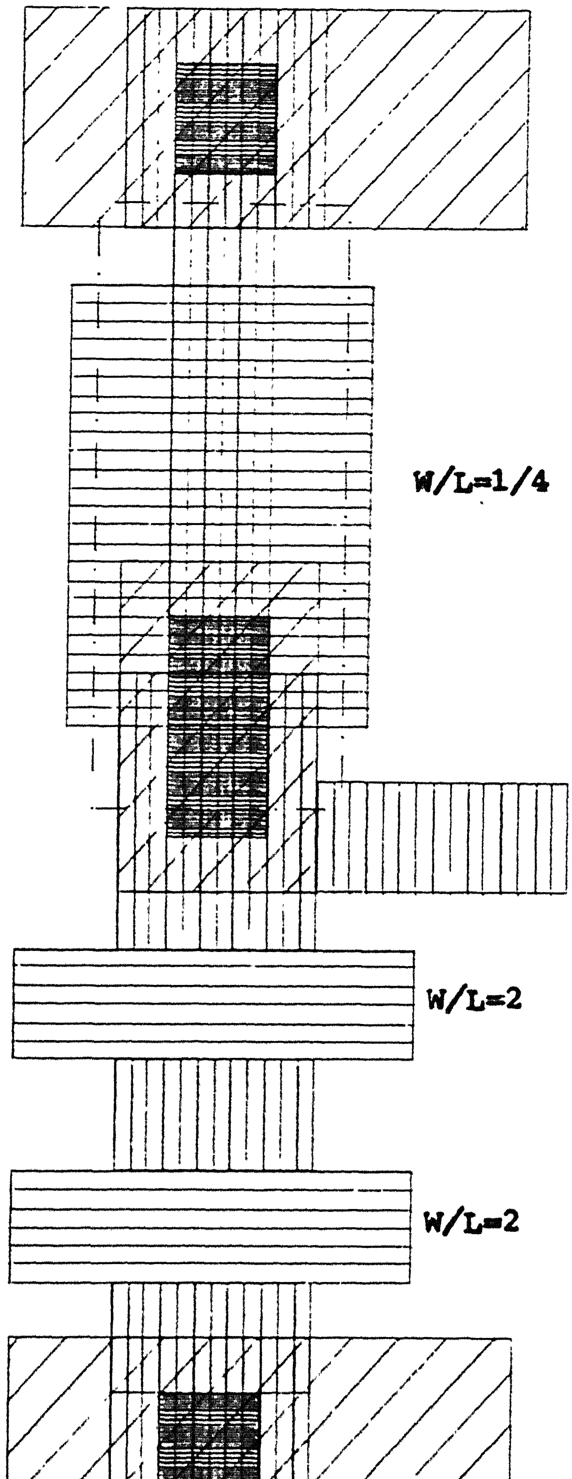


Fig. 2.9 Layout of a 2-input NAND Gate

(1) Random Logic, which is the form of logic arrangement using individual NOR gates, NAND gates and inverters,

(2) Transistor Switch Arrays, where transistors are used as switches to control the flow of logic from input to output of the combinational logic,

(3) Distributed input gate structures, which represent a structured form of random logic design and offer improved versatility and simplicity of design in comparison to random logic,

(4) Programmable logic arrays, a generalized and highly structured design architecture for the realization of both combinational and sequential logic design.

2.6 The Programmable Logic Array

There is a way to map irregular combinational and sequential functions on to regular structures, using the programmable logic array (PLA) as described below. This technique has a great advantage: functions may be significantly changed without requiring major changes either in the design or the layout of the PLA structure.

One very general and regular way to implement a combinational logic function of n-inputs and m-outputs is to use a memory of 2^n words of m-bits each. The n-inputs form an address into the memory and the m-outputs are the data contained

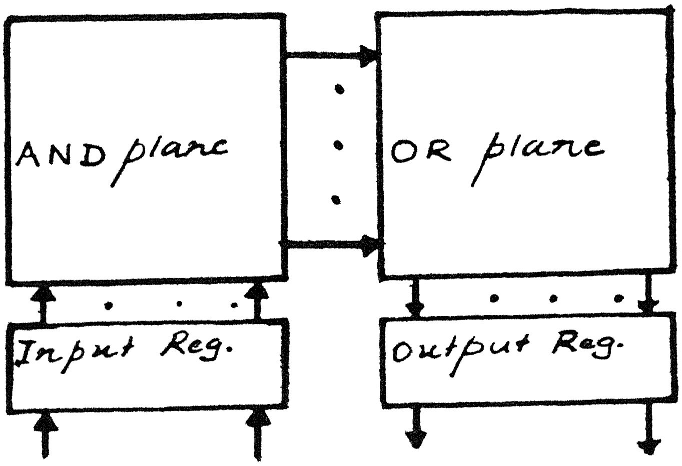


Fig. 2.10 Over-all Structure of the PLA

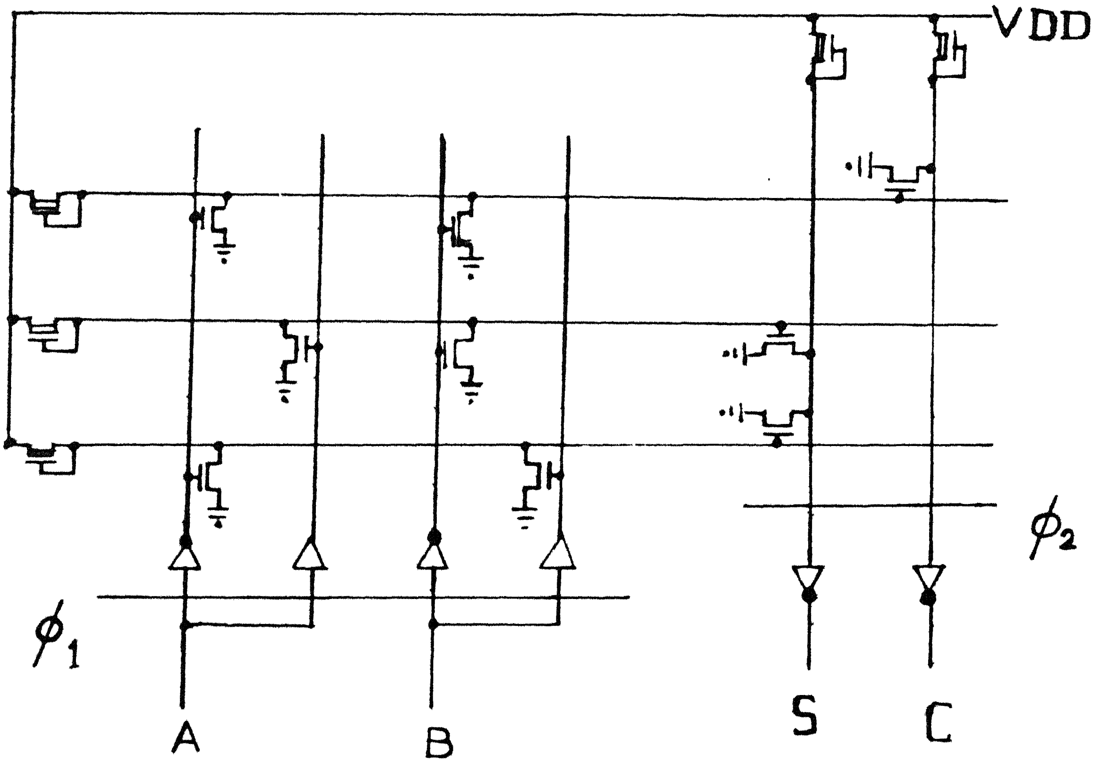


Fig. 2.11 PLA-Circuit Diagram of the Half Adder

in that address. Such a memory implements the full truth table for the output functions. A common form of memory for this purpose is the read-only memory (ROM) where the data bits are permanently placed in the memory either by a mask pattern or by electrically altering the individual bit positions. But, due to the nature of the problem, most of the possible input combinations seldom occur. In other words, many logic functions require only a small fraction of all 2^n product minterms for a canonical sum of products implementation. In such cases, a ROM consumes more area.

The PLA structure need contain a row of circuit elements only for each of those product terms that are actually required to implement a given logic function. Since it does not contain entries for all possible minterms, it is usually far more compact than a ROM implementation of the same function. To achieve full compaction, the various output functions must be jointly minimized before the PLA layout pattern can be defined.

Fig. 2.10 illustrates the overall structure of a PLA. The diagram includes the input and output registers. The inputs stored during the phase $\emptyset_1$ in the input register, are run vertically through a matrix of circuit elements called the AND plane. The AND plane generates specific logic combinations of the inputs and their complements. The outputs of the AND plane leave at right angles to its inputs and run horizontally through another

matrix called the OR plane. The outputs of the OR plane then run vertically and are stored in the output register during the phase $\emptyset_2$.

The circuit diagram of a specific PLA Half-adder **shown** in figure 2.11, will help to clarify the structure and function of the AND/OR planes of the PLA. The input register bit for each input path is formed by a pass transistor clocked on $\emptyset_1$ leading to both inverting and non-inverting super buffers. The buffers drive two lines running vertically through AND plane, one for the input term and one for its complement. The outputs of the AND plane are formed by horizontal lines with pull-up transistors at their left most end. The function of the PLA's AND plane is then determined by the locations and the gate connections of the pull-down transistors connecting the horizontal lines to ground.

Each output running horizontally from the AND plane carries the NOR combination of all input signals that lead to the gates of the transistors attached to it. For example, the horizontal row labelled R3 has two transistors in the AND plane one controlled by A', one by B'. If any of these inputs are high, then R3 will be pulled down towards ground and will be low. Thus,

$$R3 = (A' + B')' = A\ B$$

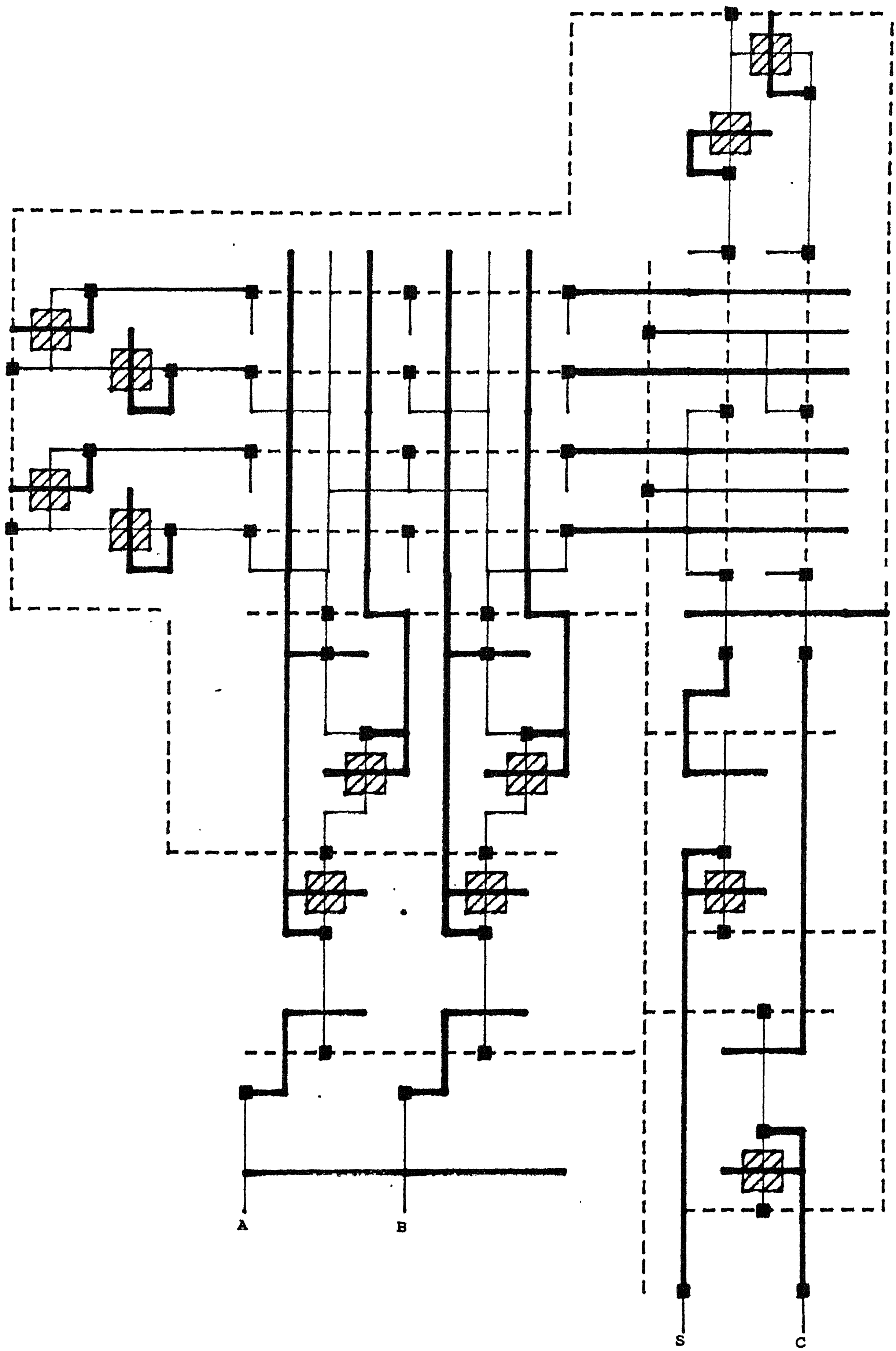
Similarly,

$$R2 = (A + B')' = A'B \quad \text{and}$$
$$R1 = (A' + B)' = AB'.$$

The OR plane matrix of circuit elements is identical in form to the AND plane matrix, but rotated by 90 degrees. Once again, each of its outputs is the NOR of the signals leading to the gates of all transistors attached to it. In figure 2.11, for example, both R1 and R2 lead to the gates of transistors leading from the output line Z_1' to the ground. If either R1 or R2 is high, Z_1' will be low. Thus $Z = \text{NOR}(R1, R2) = (AB' + A'B)'$. Up to this point, the PLA implements the NOR-NOR canonical form of Boolean function of its inputs.

The output lines of the OR plane matrix are run into an output register formed by pass transistors (located on $\emptyset_2$) leading into inverting drives. The output at this point is $Z_1 = AB' + A'B$. This expression illustrates why the two PLA planes, each implementing the NOR function, are usually referred to as the AND plane and the OR plane. Following the output register, the outputs appear directly as the sum of product form of Boolean functions of the PLA inputs, that is, as the OR of AND terms. Each horizontal line of the PLA carries one product term.

Fig. 2.12 shows one possible layout topology (stick diagram) for implementing the PLA in MOS circuitry. The


A
B
S
C

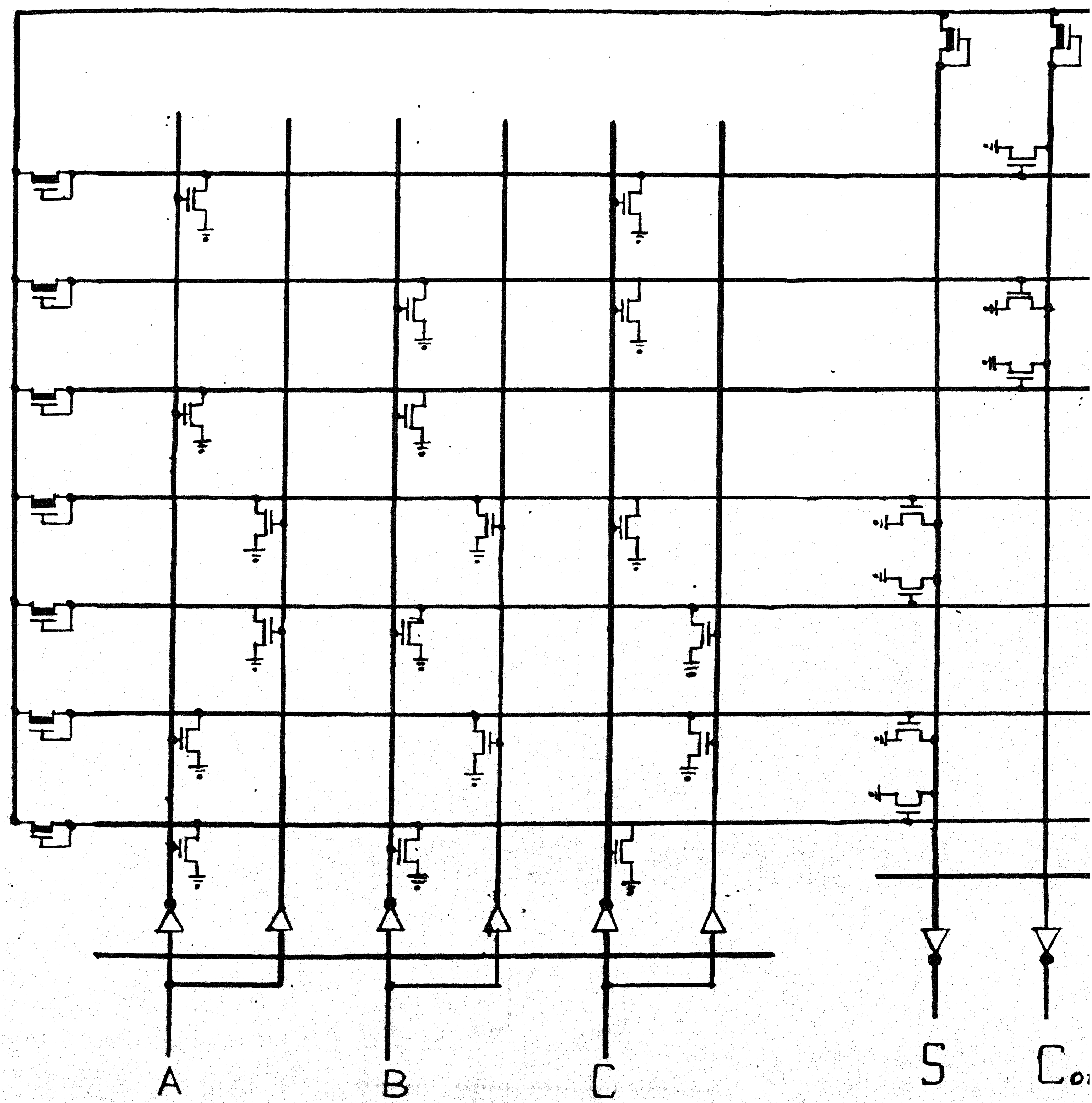
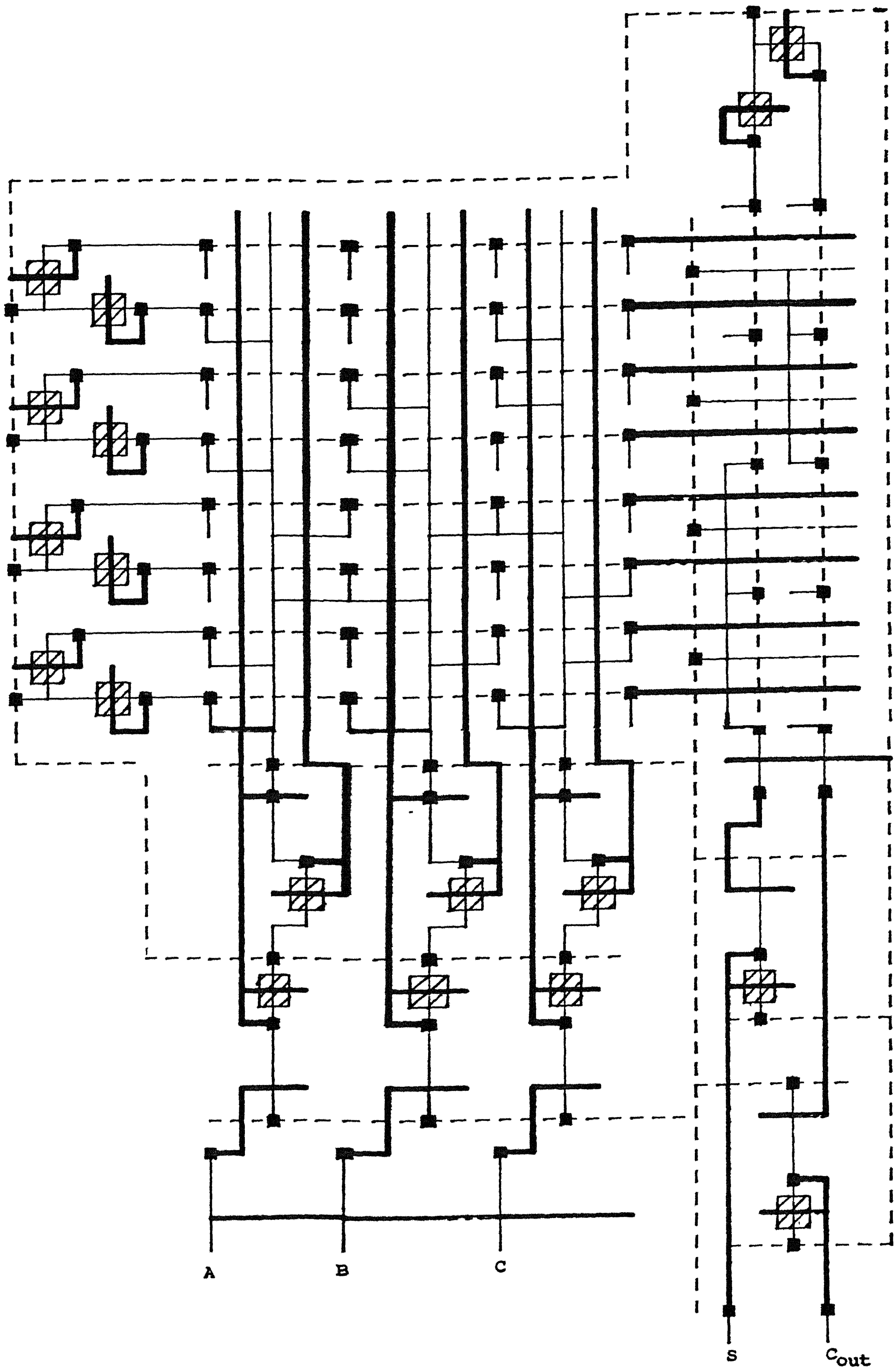


Fig. 2.13 PLA-Circuit Diagram of the Full Adder


A
B
C
S
C_{out}

example is the Half-adder. The input lines (thick lines) crossing each plane run in poly. The output lines from each plane are run in metal (dashed lines). Paths running to ground are placed between alternate poly lines on the diffusion level (thin lines). It is then a simple matter to form the pull down transistors to be connected between metal output lines and the ground. They are selectively located diffusion lines under the appropriate input poly lines. The circuit diagram and stick diagram of the Full adder are shown in Fig. 2.13 and 2.14.

Although, PLA may implement a very regular structure, the irregularity is confined to irregular locations of pull-down transistors, that 'program' the function. The overall shape and size is a function of the parameters :

(1) the number of inputs,

(2) the number of product terms,

(3) the number of outputs, and

(4) the minimum feature size (2λ).

2.7 The PLA Parameters

The driver transistor and the load transistor of the PLA layout and **stick diagram** have been designed with the following aspect ratios :

<u>Driver</u> : $W = 2\lambda$; $L = 2\lambda$ $(\beta_{driver} = 1)$

<u>Load</u> : $W = 2\lambda$, $L = 4\lambda$ $(\beta_{load} = 1/4)$

The parameters, area, power dissipation and time delay are calculated as follows :

1. Area of PLA Layout

Length = $(16\lambda \times \text{No. of inputs}) + 16\lambda \times \text{Roof (No. of outputs}/2 + 49\lambda)$

Height = $(16\lambda \times \text{Roof (No. of prod. terms}/2) + 88\lambda)$

Area = Length x Height

2. Power Dissipation

A NOR gate dissipates maximum power when all inputs are high. However, the difference is negligible and the power dissipated can be approximated as,

$$\text{Power} = (\text{No. of outputs} + \text{No. of product terms}) \times \text{VDD} \times I_o$$

where $I_o = \frac{\mu C_o}{2} (W/L) \text{ load} \times V_{th}^2 (\text{load})$

3. Time delay

The total delay of an inverter pair is given by,

$$\tau = (k + 1)\ \tau$$

where $k = (W/L)\ \text{load}/(W/L)\ \text{driver}$ and

$$\tau = L^2/\mu V_{ds}$$

when paragitic effects are neglected.

For the NOR plane C_{total} is,

$$\text{Delay} = \frac{C_{total}}{C_g} (k + 1)\ \tau$$

where

$$C_{total} = \text{gate capacitance} + \text{stray capacitance} + 2(\text{Miller capacitance})$$

CHAPTER 3

DESCRIPTION OF THE PACKAGES

3.1 NMOS Design Rules

Five types of regions are to be identified in the layout and stick diagram. They are,

(1) Diffusion region,

(2) Polysilicon region,

(3) Ion implantation region,

(4) Metalization region and

(5) Contact cut region.

The following convention has been used to represent the different regions in the layout diagram.

Diffusion region	: Rectangular box with vertical lines, as shown in fig. 3.1(a).
Polysilicon region	: Rectangular box with horizontal lines as shown in fig. 3.1(b).
Ion implantation region	: Rectangular box formed of alternating dashes and dots as shown in fig. 3.1(c).
Metalization region	: Rectangular box with lines inclined at 45 degrees to horizontal axis, as shown in fig. 3.1(d).

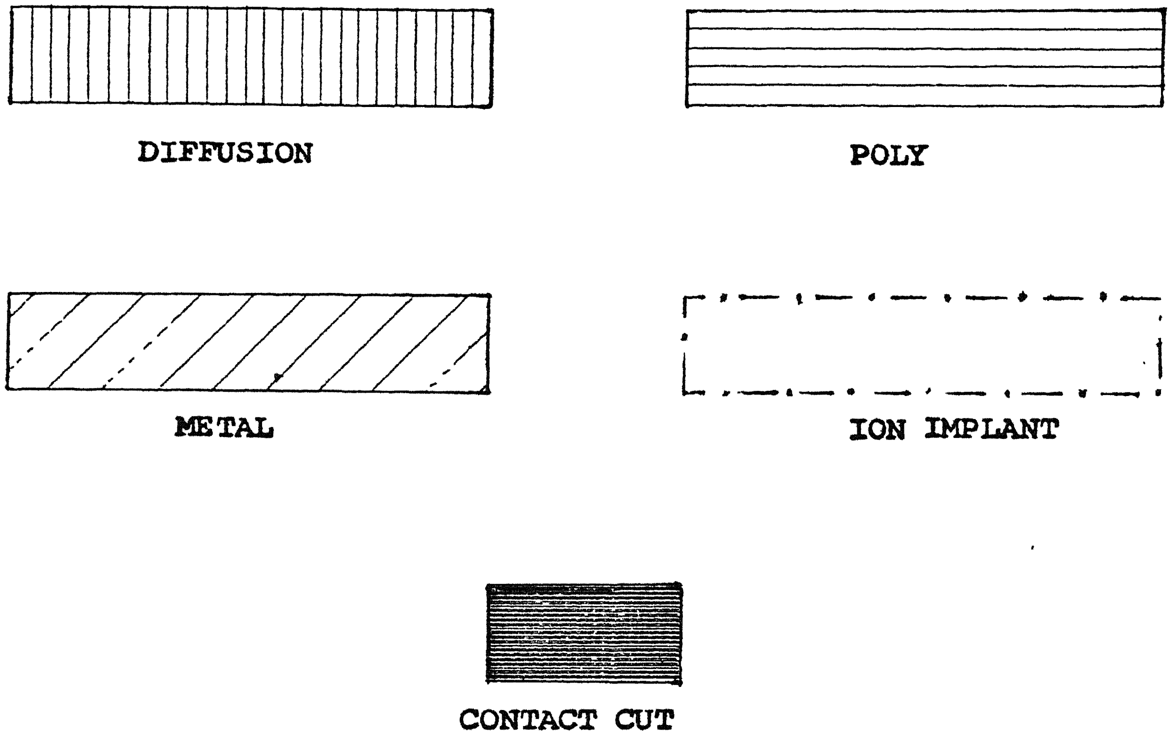


Fig. 3.1

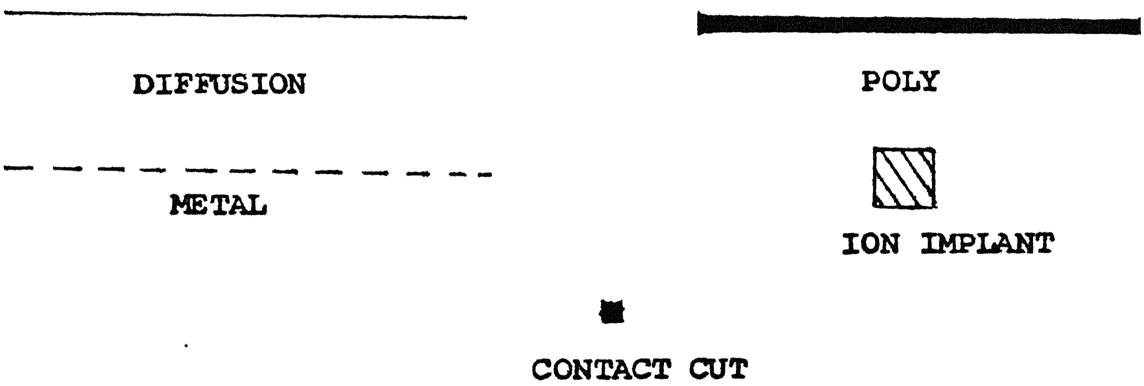


Fig. 3.2

Fig. 3.1 Representation of Different Regions in Layout

Fig. 3.2 Representation of Different Regions in Stick Diagram

Contact cut region : Contact cuts are used to connect

(a) polysilicon region and diffusion region,

(b) Diffusion region and metalization region and

(c) Polysilicon region and metalization region.

The contact cut is represented by a blackned box (Fig.3.1(e)).

The minimum distance of separation between various regions is depicted in Fig. 3.3.

The following convention applies for the representation of different regions in stick-diagram.

Diffusion (region)	: continuous thin line
Polysilicon (region)	: thick line
Metalization	: line formed of dashes
Ion implantation	: rectangular box with 45 degrees lines
Cut	: 2mm x 2mm blackened rectangular box

The representation of different regions in the layout and the stick diagram are shown in Fig. 3.1 and Fig. 3.2 respectively.

3.2 Computer Aids For Random Logic Implementation

This package has basic building blocks like Inverter, NAND gate, NOR gate and descriptions of the shapes of the device layers (poly, diffusion, ion-implantation, etc.), on the device library.

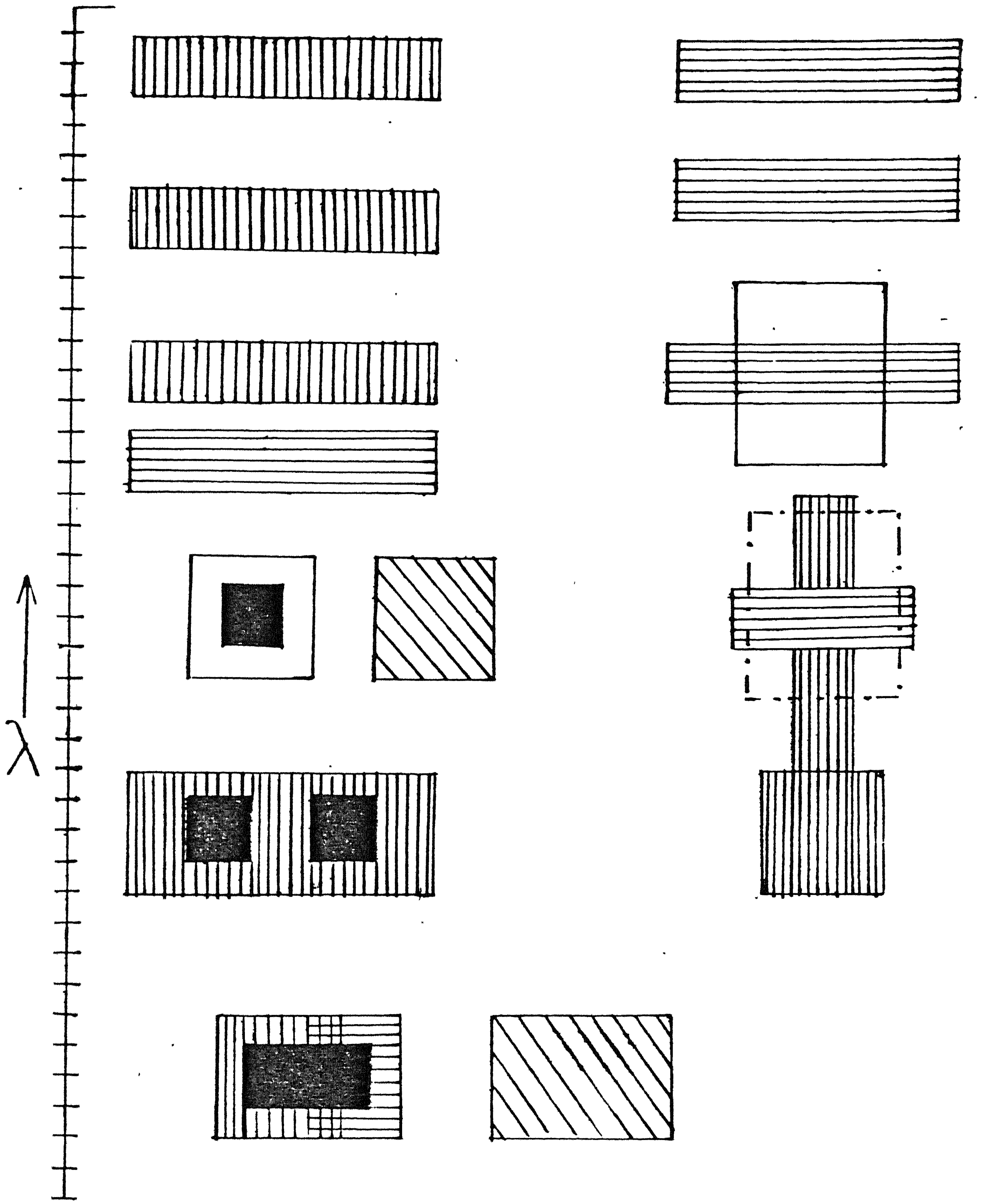


Fig. 3.3 NMOS Design Rules of Layout

The topology of the standard gates was manually created by drawing shapes, polygon-by-polygon. Later on the layout was digitized point by point and entered into the memory.

If the system whose layout is desired, has a few cell types that are replicated over andover, plus a little random wiring, only a single copy of each cell type is drawn. The replication of cells in various orientations and locations in the system layout can then be easily described using any graphics language. The regularity of the design governs the ease with which a system's layout can be generated in this way.

The subroutine 'BOX' - a part of this package, is similar to a macroassembler. It can be used to describe the layout of device layers. A basic cell is then built as a collection of layers of different shapes and sizes one above the other. These basic cells can then be placed at the desired locations with desired orientations. A bit of random wiring on the layout obtained so far, completes the layout of the system.

The appendix - A, describes the various subroutines available in this package. Fig. 3.4 shows the layout of a two-bit shift register. This is easily created by using the layout of the inverter and pass transistor, and some random wiring.

3.3 PLA Stick Diagram Generator

The outline of the stick diagram shown in Fig. 3.5, is constructed from the following five cell pair structures.

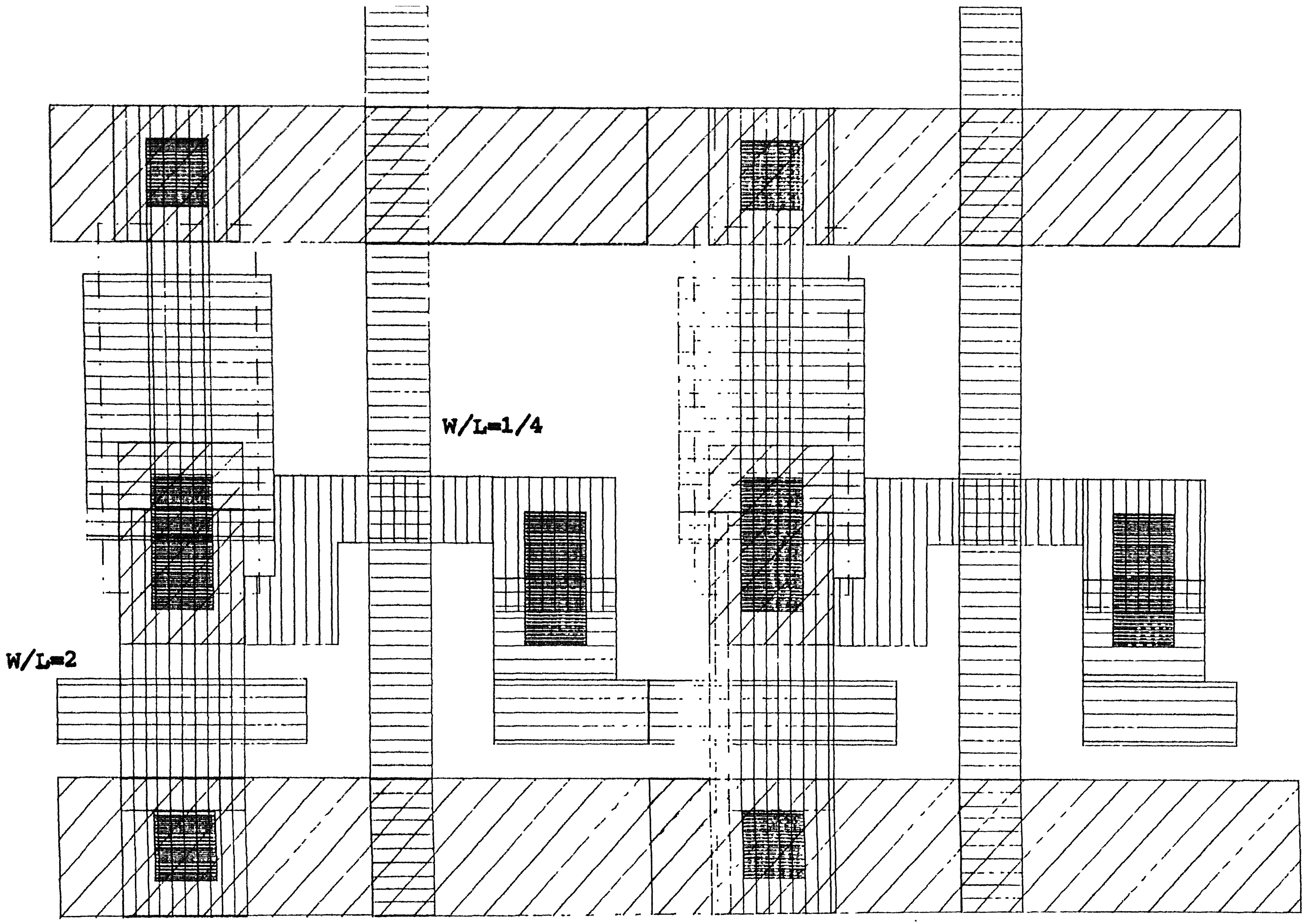


Fig. 3.4 Layout of a 2-bit Shift Register

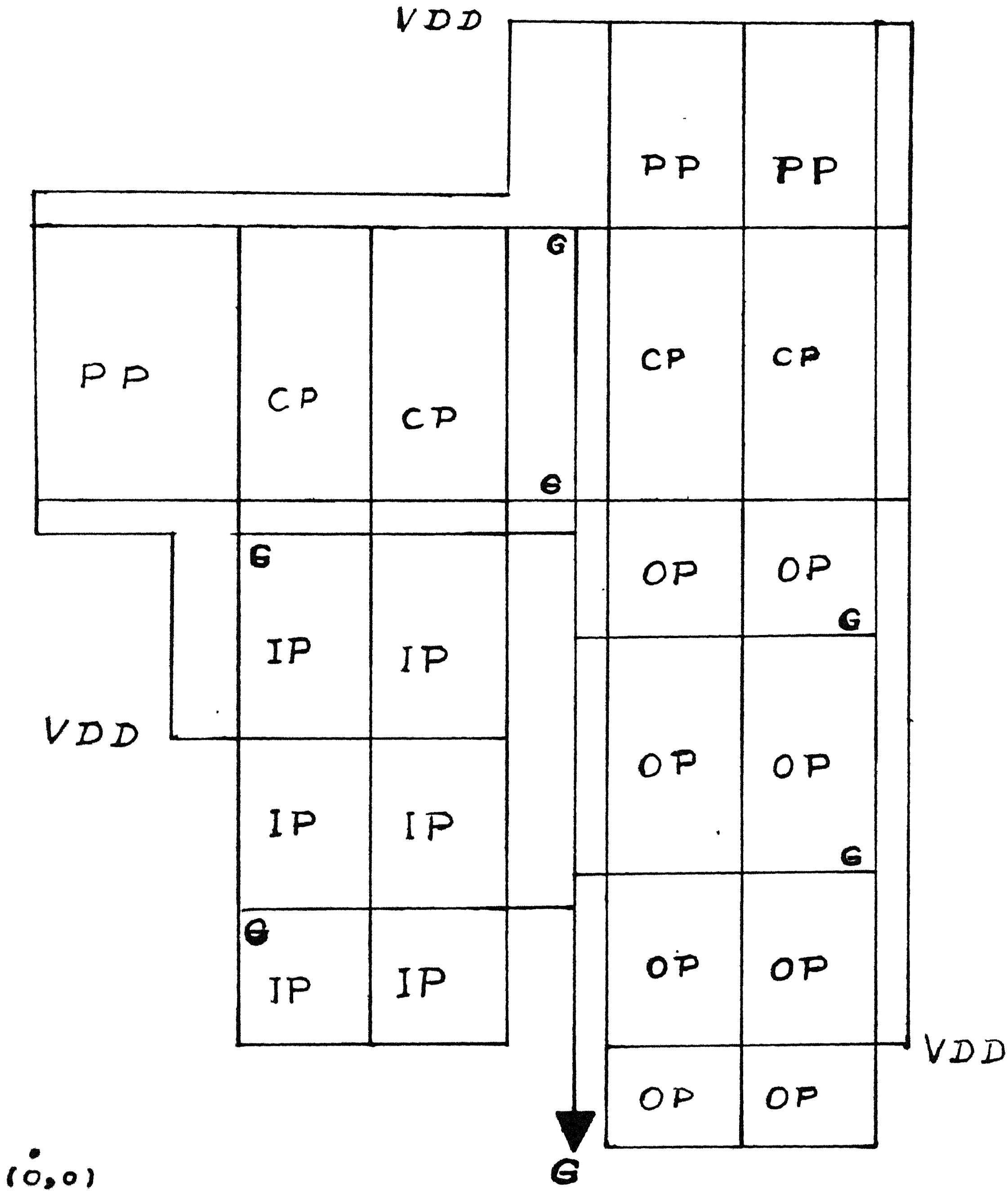


Fig. 3.5 Block Schematic of the Stick Diagram

(1) cell pair (for the AND and the OR planes),

(2) pull-up pair (for the AND and the OR planes),

(3) AND-OR connect,

(4) Input driver pair and,

(5) output driver pair.

The total number of cell pairs along the x-y axes, in the OR and AND planes are given by,

No. of cell pairs along the x-axis in the AND plane } = No. of inputs,

No. of cell pair along the y-axis (in AND and OR planes } = Roof (No. of product terms/2) and

No. of cell pairs along the x-axis in the OR plane } = Roof (No. of outputs/2).

The basic stick diagram structure is constructed from the number of inputs, number of product terms, number of outputs, the x and y co-ordinates of the lower left corner of the stick-diagram and the minimum feature size factor lambda. It is then programmed by placing the pull down transistors in the AND/OR planes.

<u>Cellpair</u> : The stick-diagram representation of the cell pair is shown in Fig. 3.6(a). It has two poly lines. The left poly line

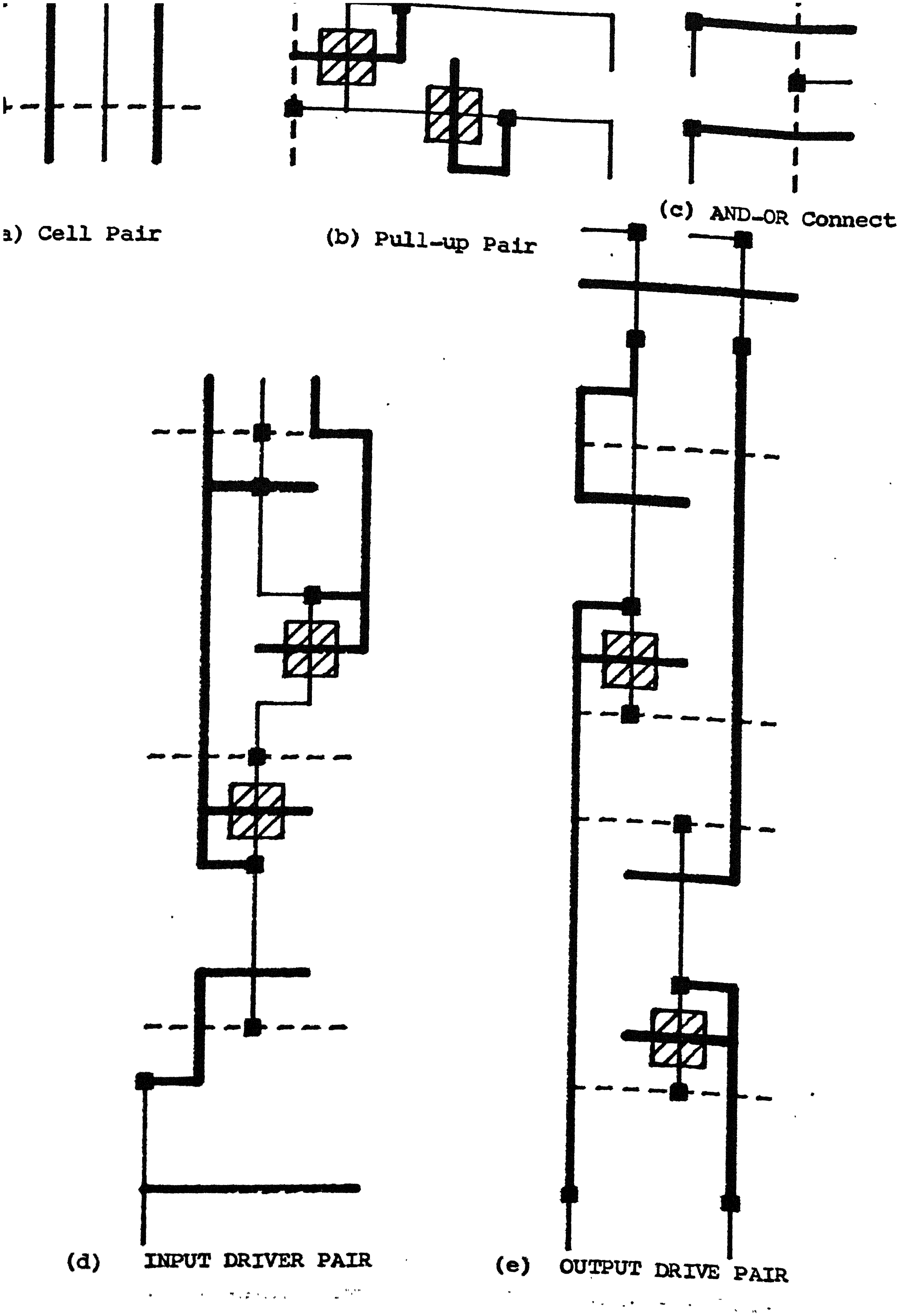


Fig. 3.6 Stick Diagram Representation of Cell-pair Structures

is fed by the inverted input and the right by the true input. The two horizontal metal lines carry the outputs of the rows (multi-input NOR gates) of the AND plane and feed the same to the OR plane via the AND-OR connect (see Fig. 3.5). The diffusion lines seen between the poly lines carry the ground connection. Two pull down transistors can be formed in this cell pair, either with the true input poly line or its complement, by forming a diffusion line so as to cross the appropriate poly line. This occupies an area of 4 'lambda' by 4 'lambda', 'Lambda' being the minimum feature size factor.

Pullup pair : The stick diagram of the pull up pair shown in Fig. 3.6(b), has a pair of depletion mode transistors. They form the load for the rows (multi-input NOR gate structure) of the AND plane. This has an area of 6 'lambda' by 4 'lambda'.

AND-OR connect : The AND-OR connect shown in Fig. 3.6(c) provides the interface between the AND and the OR planes. The two poly lines running horizontally in this cell structure help to connect the metal lines of the AND plane to the poly lines of the OR plane.

Input driver pair : This provides both true input and its complemented form. The input driver pair gets the inputs via the pass transistors clocked on the phase $\emptyset_1$. The lower inverter

provides complemented form of the input, whereas the upper one provides the true form. The stick diagram of the input driver pair is shown in Fig. 3.6(d).

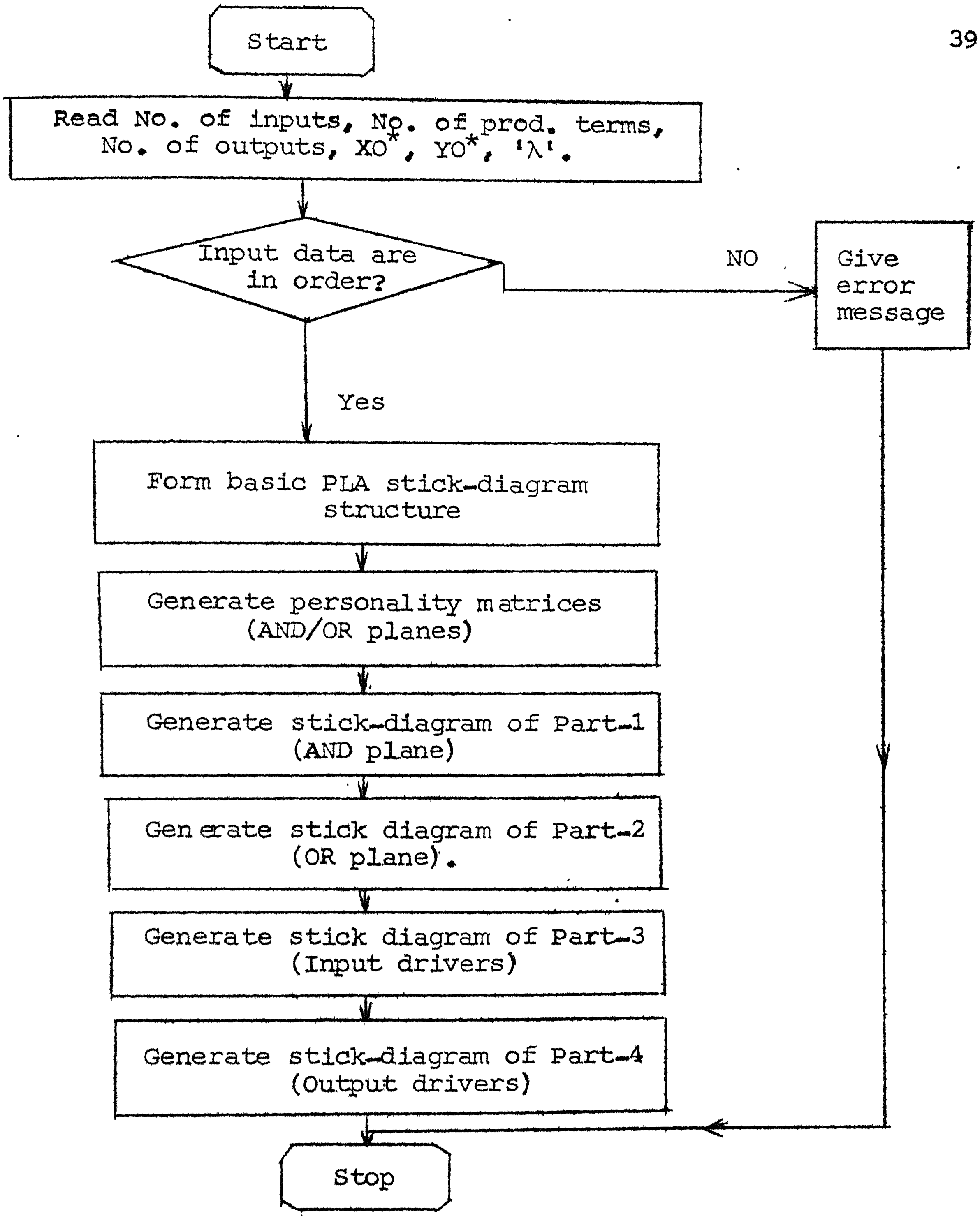
Output driver pair : The stick diagram of this structure shown in Fig. 3.6(e), has two inverters, one for each output of the OR plane. The outputs of the OR plane are passed on to the output driver pair via the pass transistors clocked on ϕ_2. This occupies an area of 4 'lambda' by 69 'lambda'.

After obtaining the basic stick diagram structure using the five cell pair structures described, programming (placing the pulldown transistors in the AND/OR planes) follows in accordance with the contents of the personality (AND/OR planes) matrices. The details of the generation of the personality matrices, interpretation of its contents, etc. are given under the PLA Layout Generator section as it is identical with the personality matrices of the PLA Layout Generator.

The flow chart of the 'Stick Diagram Generator' program is shown on the next page.

3.4 PLA Layout Generator

Six types of cell pair structures are needed to construct the basic structure of the PLA layout. They are ,



Flow chart for PLA Stick-diagram Generation

* : XO, YO are the x and y co-ordinates of the lower left corner of the PLA stick-diagram

(a) PLA cellpair,

(b) PLA pullup pair,

(c) PLA connect,

(d) PLA Ground,

(e) PLA input driver pair and

(f) PLA output driver pair

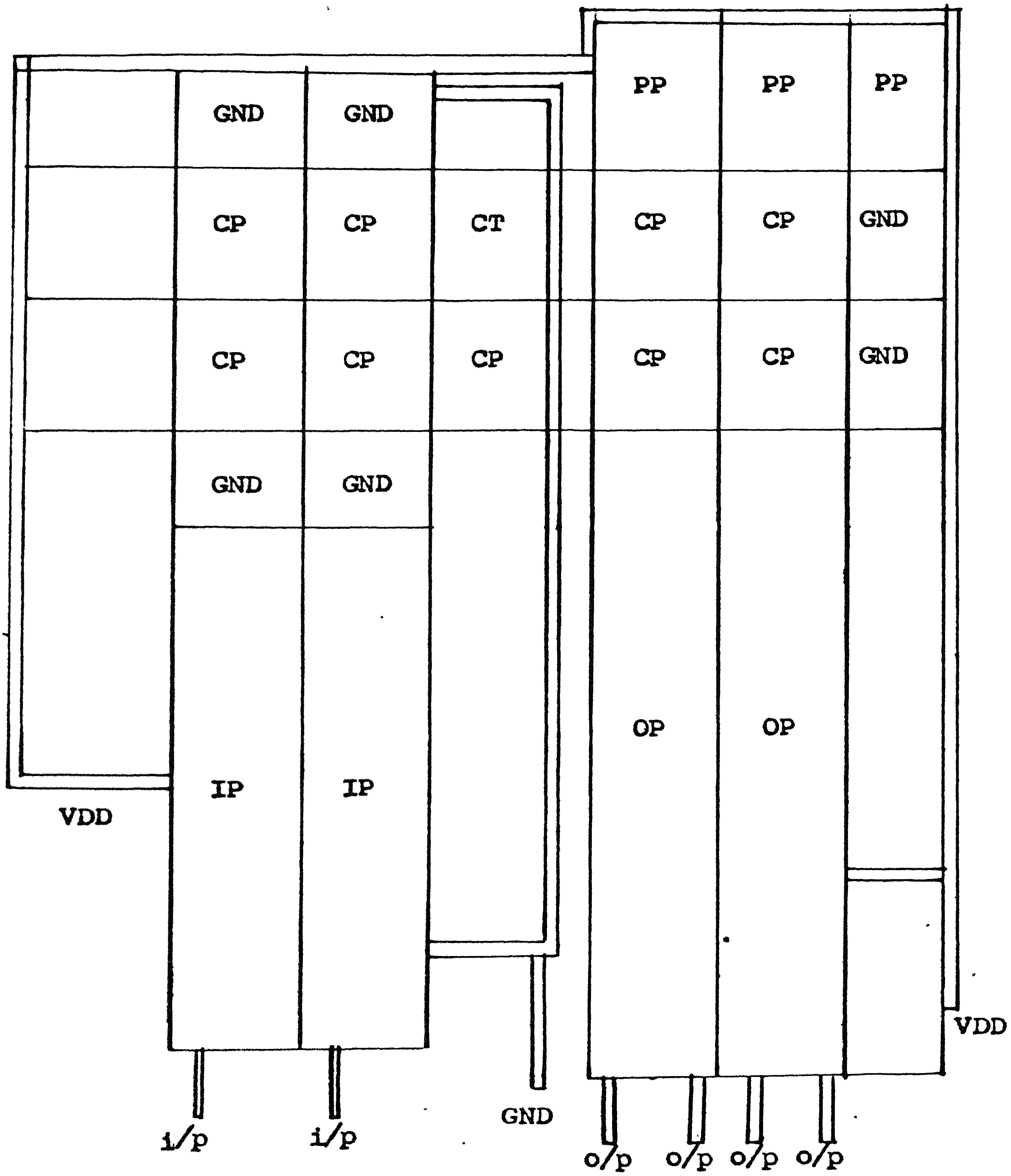
The outline of the basic structure of the PLA using these six types of cell pair structures and metal interconnections is shown in Fig. 3.7. The PLA structure is first constructed according to the number of inputs, the number of product terms, the number of outputs, and the minimum feature size factor 'lambda'. This structure will have,

No. of cell pairs in the y-direction = Roof (No. of product terms/2),

No. of cell pairs in the x-direction (AND plane) = No. of inputs and

No. of cell pairs in the n-direction (OR plane) = Roof (No. of outputs/2).

It is programmed by placing the pulldown transistors in the AND and OR planes. The locations of the pulldowns are obtained by the personality matrix (described later).



CP=Cellpair ; CT=AND-OR Connect; PP=Pull-up pair
GND=Ground ; IP=Input Driver pair ; OP=Output Driver Pair

Fig. 3.7 Block Schematic of the PLA Layout

The layout description of the basic cell pairs is given below.

Cellpair : The layout of the cellpair for the AND plane is shown in Fig. 3.8(a). It has two horizontal layers of metalization, two vertical layers of polysilicon, and a vertical diffussion layer betweeen the two polysilicon layers. This diffusion layer finally terminates at the ground connection. A contact cut region is seen at the left end of each of the metal layers. Two pulldown transistors can be formed in this cell pair either with the true input or its complement. This is done by forming a diffusion layer so as to cross either of the poly gate for the true input or its complement. This cellpair occupies an area of 14 'lambda' by 14 'lambda', 'lambda' being the minimum feature size factor.

Pullup pair : The layout of this cell pair is shown in Fig. 3.8(b). It has two depletion load transistors placed one above the other. Each depletion transistor forms the load for the NOR gate structure of one row of the AND plane. It occupies an area of 21.5 'lambda' by 14 'lambda'.

PLA Connect : The PLA connect shown in Fig. 3.8(c), is used to provide the interface between the AND plane and the OR plane of the PLA. The two horizontal poly layers of this cell pair connect the AND plane metal layers to the OR plane poly lines.

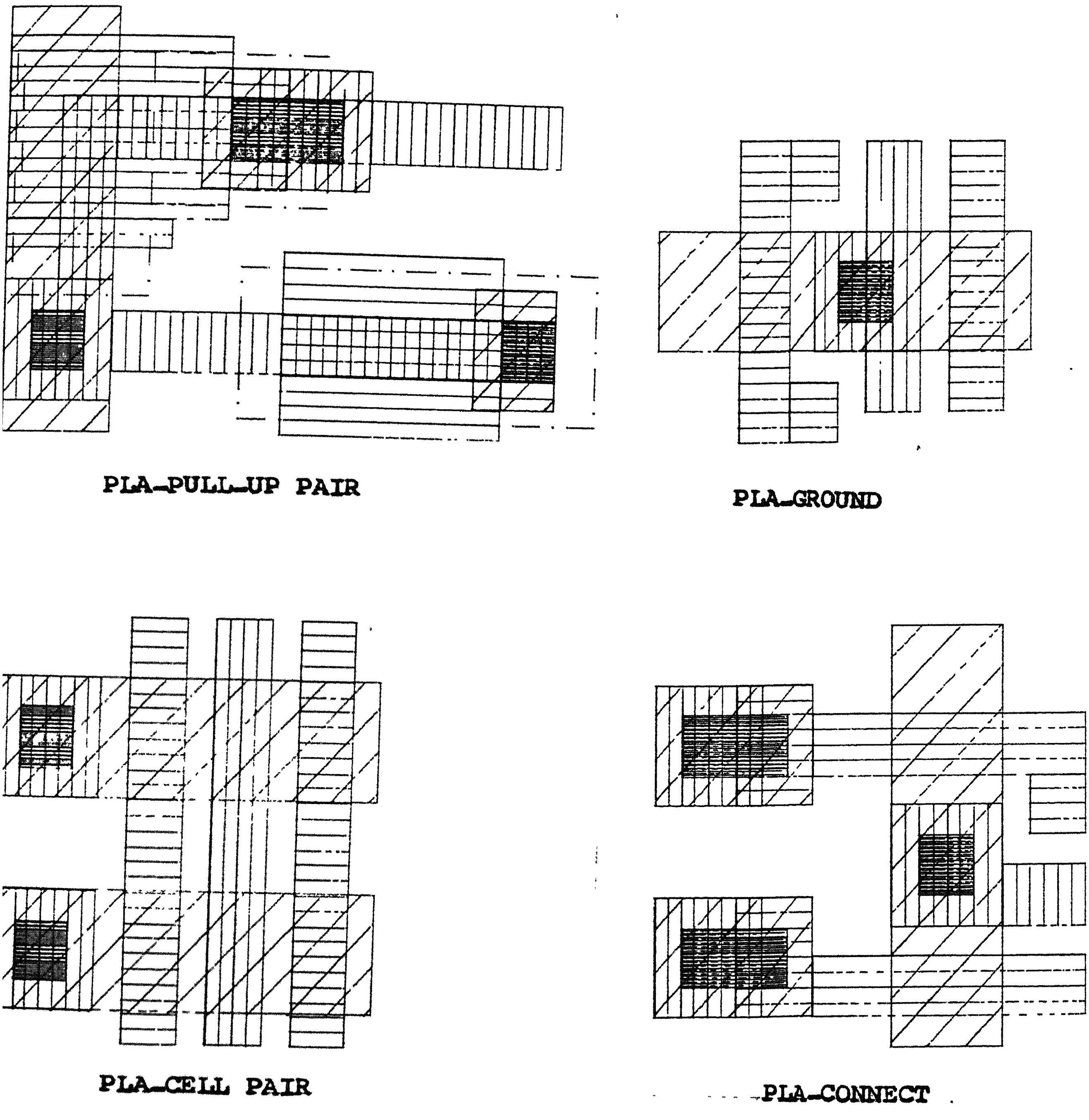


Fig. 3.8 Layout of the Cell-pair Structures

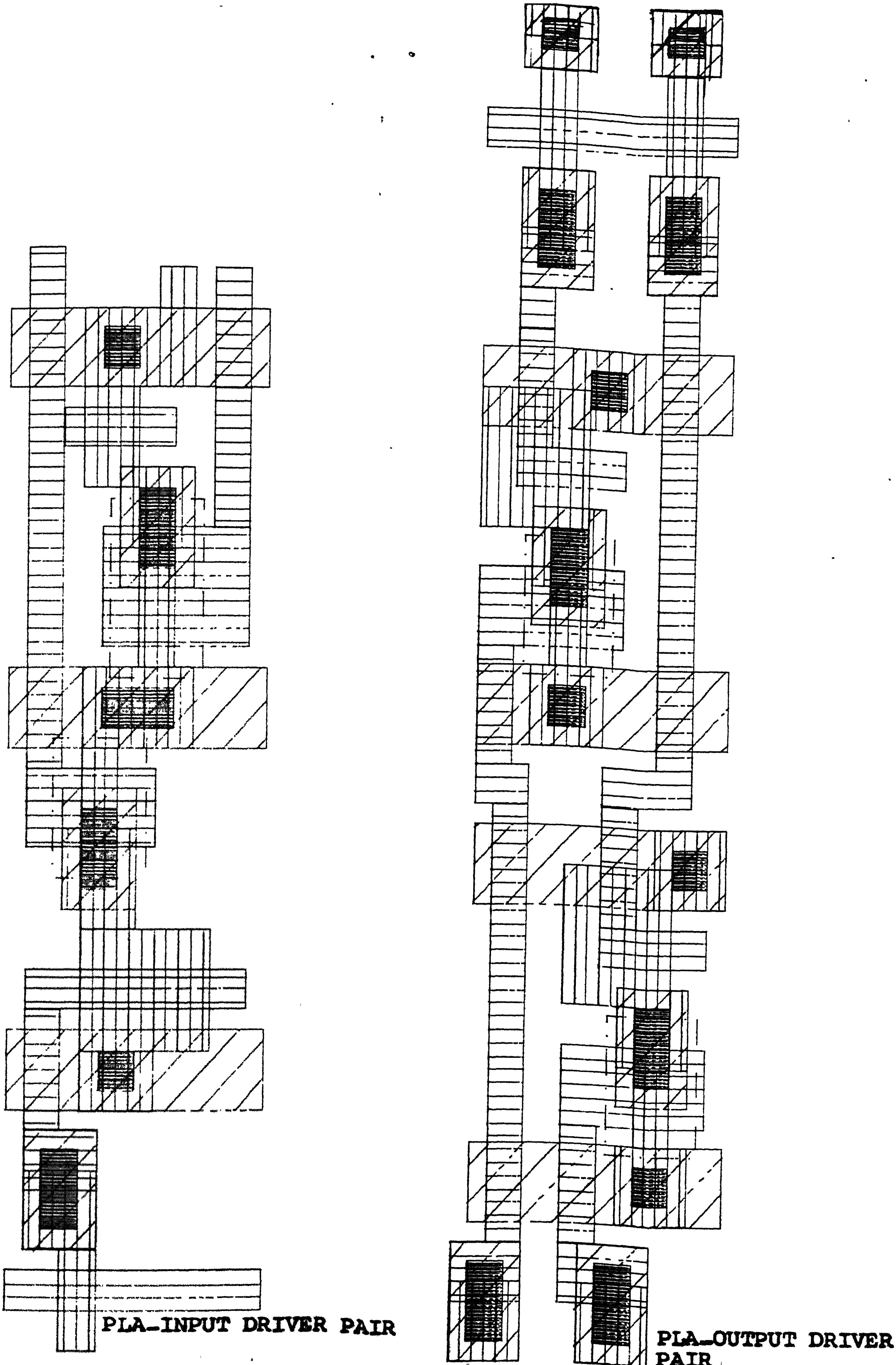


Fig. 3.8 Layout of the Cell-pair Structures

The vertical metal layer carries the ground connection. The diffusion layer between the two polysilicon layers of this cell pair joins the horizontal diffusion layer running between the poly layers of the OR plane. The PLA connect has an area of 16 'lambda' by 16 'lambda'.

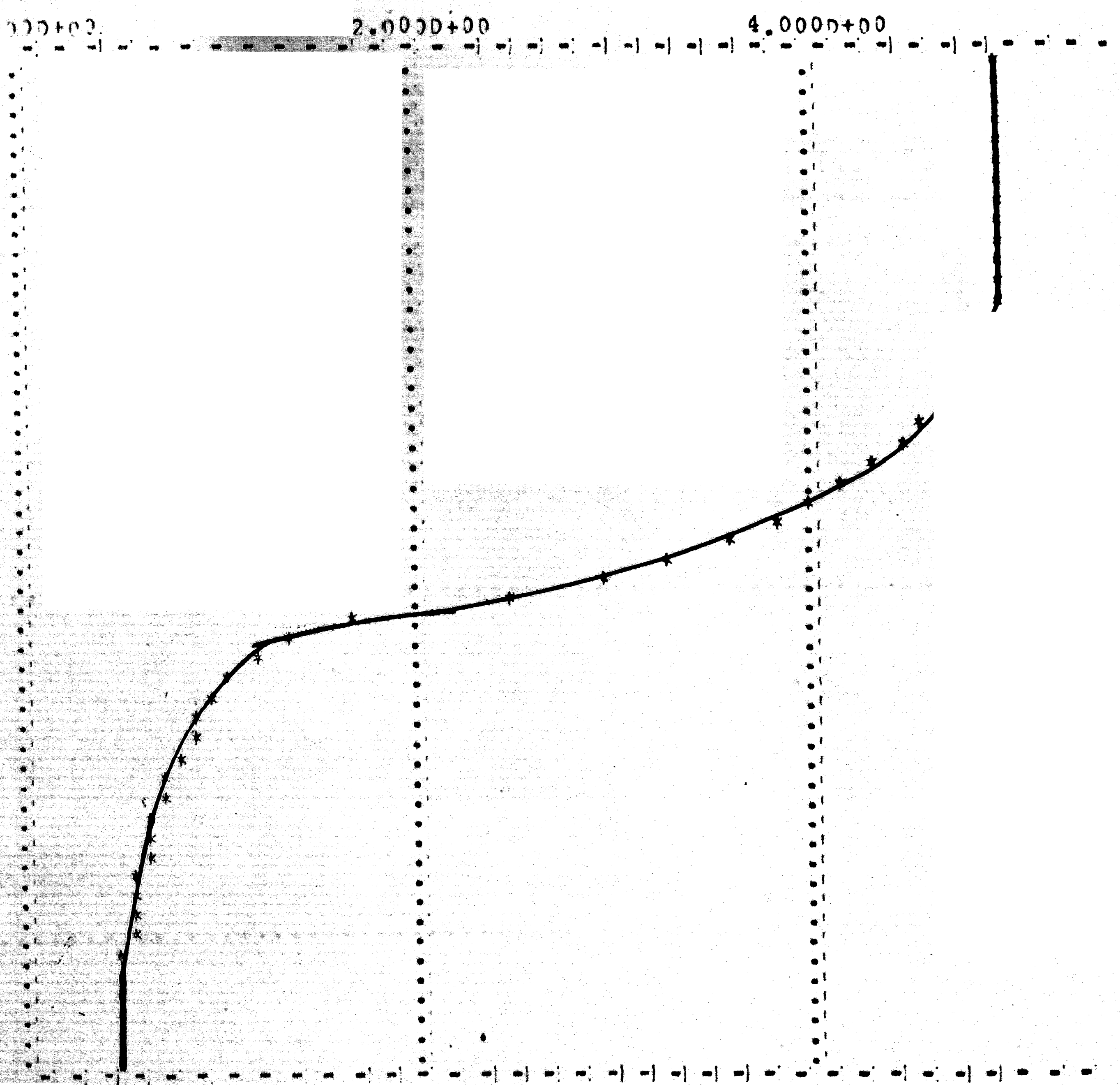
PLA Ground : Fig. 3.8(d) is the layout of the PLA Ground of the AND plane. Two vertical polysilicon layers seen in Fig. 3.8(d), connect the inverted and the true inputs to the AND plane multi-input NOR gates. The diffusion line of the ground cell connects the ground point to the diffusion (vertical) of the AND plane. This spreads over an area of 14 'lambda' by 10 'lambda'.

Input Drive pair : This is formed of two inverters, the output of one going to the other. The output of the lower inverter provides the complemented input, whereas the output of the upper one provides the true input. The layout of this cell pair is shown in Fig. 3.8(e). This occupies an area of 14 'lambda' by 55 'lambda'. The pass transistors clocked on the phase ϕ_1 allow the inputs to the PLA to appear at the inputs of the driver pair.

The cell pair, pullup pair and the ground cell of the OR plane are similar to those of the AND plane except for a rotation of 90 degrees in the clockwise direction.

The output driver pair connected to the OR plane output has a pair of invertersone for each of the outputs of the OR

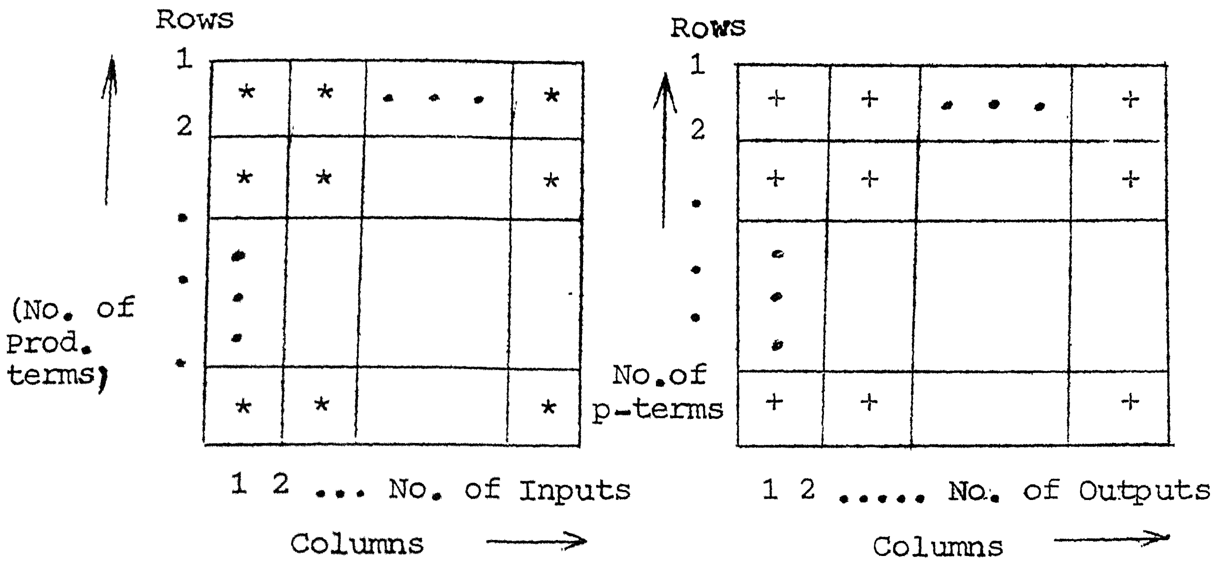
0.0000D+00	5.0000D+00
1.0000D-01	5.0000D+00
2.0000D-01	5.0000D+00
3.0000D-01	5.0000D+00
4.0000D-01	5.0000D+00
5.0000D-01	5.0000D+00
6.0000D-01	5.0000D+00
7.0000D-01	5.0000D+00
8.0000D-01	5.0000D+00
9.0000D-01	5.0000D+00
1.0000D+00	5.0000D+00
1.1000D+00	4.9914D+00
1.2000D+00	4.9756D+00
1.3000D+00	4.9453D+00
1.4000D+00	4.9033D+00
1.5000D+00	4.8417D+00
1.6000D+00	4.7792D+00
1.7000D+00	4.6941D+00
1.8000D+00	4.5356D+00
1.9000D+00	4.4830D+00
2.0000D+00	4.3170D+00
2.1000D+00	4.1933D+00
2.2000D+00	4.0153D+00
2.3000D+00	3.8393D+00
2.4000D+00	3.5630D+00
2.5000D+00	3.2910D+00
2.6000D+00	2.9230D+00
2.7000D+00	2.4530D+00
2.8000D+00	1.7022D+00
2.9000D+00	1.3530D+00
3.0000D+00	1.1790D+00
3.1000D+00	1.0510D+00
3.2000D+00	9.7220D-01
3.3000D+00	9.0110D-01
3.4000D+00	8.4220D-01
3.5000D+00	7.9220D-01
3.6000D+00	7.4890D-01
3.7000D+00	7.1090D-01
3.8000D+00	6.7710D-01
3.9000D+00	6.4590D-01
4.0000D+00	6.1950D-01
4.1000D+00	5.9470D-01
4.2000D+00	5.7190D-01
4.3000D+00	5.5110D-01
4.4000D+00	5.3130D-01
4.5000D+00	5.1390D-01
4.6000D+00	4.9730D-01
4.7000D+00	4.8180D-01
4.8000D+00	4.6730D-01
4.9000D+00	4.5370D-01
5.0000D+00	4.4100D-01



Y

Fig. 4.1 DC Transfer Curve of an Inverter-positive going pulse

The two matrices appear as shown in Fig. 3.9(a) and 3.9(b).



Note - * = 0 or 1 or 2

\+ = 0 or 1

Fig. 3.9 General form of Personality Matrix
(a) AND plane (b) OR plane

For example, the personality matrices (AND and OR planes) of the Full Adder Circuit implemented from the logic functions viz., $S = ABC + AB'C' + A'BC' + A'B'C$ and $C_{out} = AB+BC+AC$, look as follows :

Rows	1	2	3
1	1	1	1
2	1	2	2
3	2	1	2
4	2	2	1
5	1	1	0
6	0	1	1
7	1	0	1

columns

Rows	1	2
1	1	
2	1	
3	1	
4	1	
5		1
6		1
7		1

columns

Fig. 3.10 Personality matrices of the Full adder (a) AND plane
(b) OR plane

The personality matrices are produced by a PASCAL program. The 'INPUT' file for this program, in the first line, has entries containing information about number of inputs, number of product terms, number of outputs, X co-ordinate and Y co-ordinate of the origin of the PLA and the minimum feature size factor (in mm.) 'lambda'. The lower left corner of the PLA is the origin for the PLA. The second line is a list of the input variables which must be of a single character. Blanks can be present between characters and if present, are ignored. Each of the lines 3 through (n+2), will have one of the 'n' product terms (i.e., the list of variables in each of the product term). Each of the lines (n+3) through (n+3+m), will have one of the 'm' outputs expressed in terms of the product terms which are to be ORed to get the particular output. The total number of lines in the file 'INPUT' is given by,

Total No. of lines = (No. of product terms+No. of outputs+2).

Line No.	Contents
1	No. of inputs, No. of p-terms, No. of outputs, X co-ordinate of the PLA origin, Y co-ordinate of the PLA origin, lambda.
2	List of input variables.
3	variable list making the 1^{st} p-term
< 4 >	<variable list making the 2^{nd} p-term >
⋮	⋮
< n+2 >	< variable list making the n^{th} p-term >
n+3	List of product term numbers forming the 1^{st} output.
< n+4 >	<List of p-term numbers forming the 2^{nd} output. >
⋮	⋮
< n+4+m >	<List of p-term numbers forming the m^{th} output. >

The following example will help to clarify the INPUT file format.

The logic function for the sum and carry of the full adder with the input carry C are given by,

$$S = ABC + AB'C' + A'BC' + A'B'C \text{ and}$$

$$C_{out} = AB+BC+AC .$$

The INPUT file looks as shown below :

Line No.	Contents
1	3 7 2 0.0 0.0 1.0
2	ABC
3	ABC
4	AB'C'
5	A'BC'
6	A'B'C
7	AB
8	BC
9	AC
10	1 2 3 4
11	5 6 7

Table 3.1 Full adder INPUT file for the PLA layout Generator

The entries in the first line of the table 3.1, namely, 3,7,2, 0.0, 0.0, and 1, indicate that there are 3 inputs, 7

product terms, 2 outputs. The PLA will be placed (the lower left corner is the reference point of the PLA structure), at x = 0.0, y = 0.0 with respect to the current origin. The minimum width of the lines in the PLA layout will be 2mm. (2x1.0). The second line indicates that the three inputs are A,B and C. The lines 3 through 9 are for the product terms ABC, AB'C', ..., AC. The lines 10 and 11 say that the first output of the PLA (i.e., the extreme left in the OR plane), is to be formed by ORing the 4 product terms, namely, 1, 2, 3 and 4 (i.e. ABC, AB'C', A'BC', A'B'C), and the second PLA output by ORing the 5th, 6th and the 7th product terms (i.e., AB, BC and AC).

The flow chart of the program generating the layout is shown in Fig. 3.11.

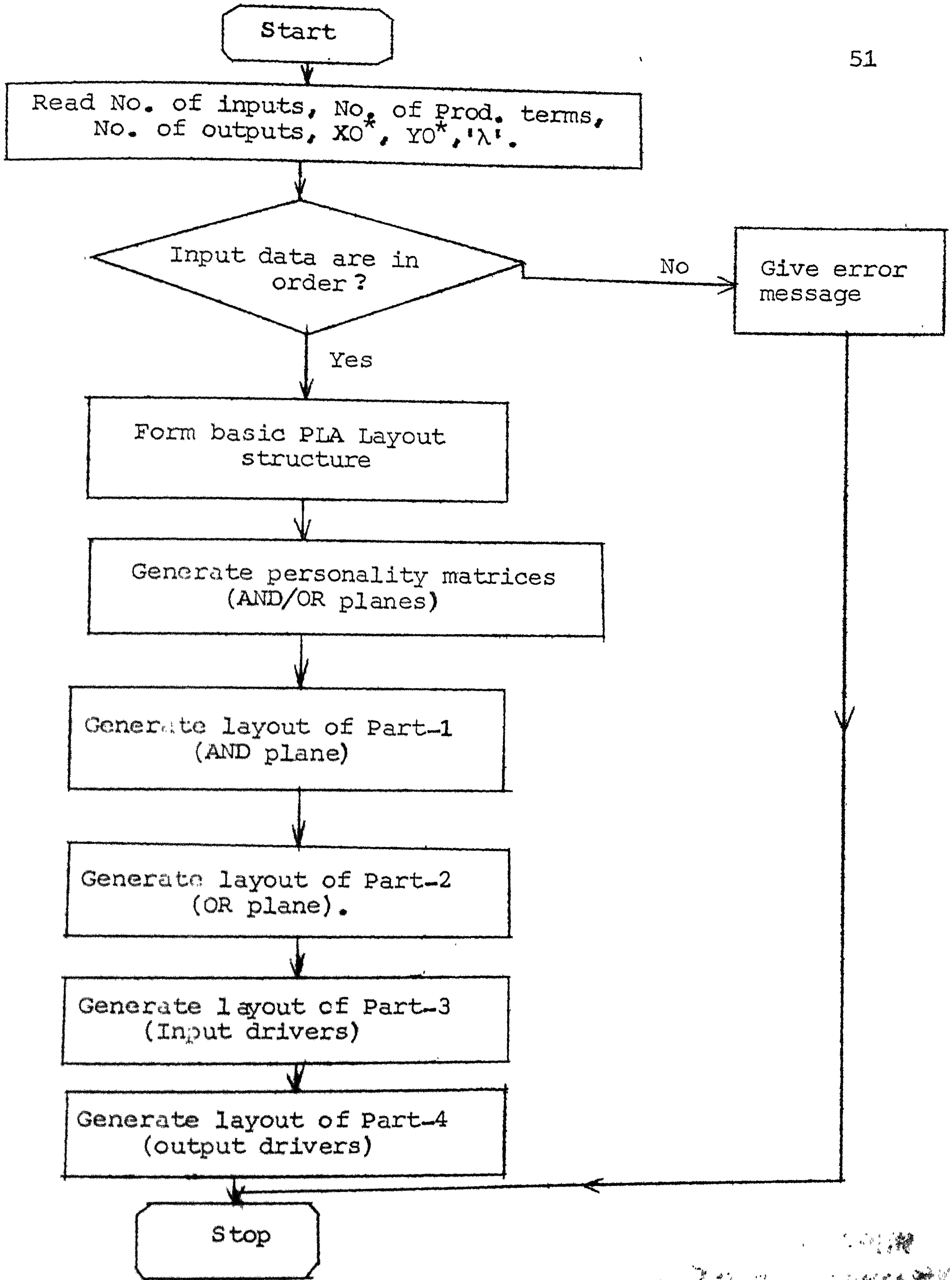


Fig. 3.11 Flow chart for PLA Layout Generation

* XO,YO are the x and y co-ordinates of the lower left corner of the PLA Layout.

CHAPTER 4

SIMULATION

4.1 Need for Simulation

If the design process is not fully automated, then the design must be checked for functional error by simulating the operation of the circuit or the system. Simulation may also be carried out as an aid in the process of building-up of the circuit to achieve the desired performance levels, particularly speed and power dissipation. Just as there may be several different levels in a design hierarchy, several different simulators may be used for each level. Ideally a simulator is required at each interface in the hierarchy which would enable a varification of the details that are created by designing the lower level from the higher level.

4.2 Circuit Simulators

At the lower end of the hierarchy, circuit simulator offer the most detailed level of simulation normally used for design. They model the electrical circuits of transistors, capacitors, etc., to determine the static and transient behaviour of node voltages and branch currents within the network. The results are usually presented in the form of plots of voltage (or current) versus time at selected nodes of the circuits, like

an oscilloscope trace. The primary input to these circuit simulators is a list of labelled circuit elements and their interconnections, including the nature of excitations at input nodes. There is also a secondary input of process parameters (threshold voltage, oxide thickness, etc.) for each type of element to be modelled. For a depletion load NMOS process, one set of parameters is used for enhancement devices, and one for the depletion devices.

The transient behaviour of the network is then modelled in a series of small time steps (say 1 nsec or less). In an elementary way, the voltages present on all of the circuit nodes at the start of the step determine the individual branch currents in the network. When multiplied by the time step these currents give net gain or loss of charge at each node which, devided by the nodal capacitance, gives the new node voltages for the real time step. The transistor currents are computed from internal models using the node voltage, transistor dimensions and process parameters given for each element. Evidently the accuracy of the simulation depends not only on these parameters, but also on the model itself. This may be a first order approximation such as the quadratic drain current relationship or it may include a range of higher order effects; sub-threshold conduction, mobility variation, etc.

4.3 Simulation Results

The SPICE deck, and the plots of the DC transfer characteristics and the transient response are given in this section.

1. DC transfer characteristics of an inverter-positive going pulse (β_R = 1:4).

Table 4.1 SPICE deck

```
C1      3      0.3 pF
VDD     4      0   DC    5V
V1      1      0   PULSE   0V   5V   30NS   20NS   20NS  1.0US  2.0US
M3      4      3     3   0   MDP   L=20U   W=5U
M1      3      1     0   0   MEN   L=5U    W=5U
.MODEL   MEN   NMOS   KP=3.0   E-5   VTO=1V
.MODEL   MDP   NMOS   KP=2.5   E-5   VTO=-4V
.DC      V1    0V     5V       0.1
.PLOT    DC    V(3)
.END
```

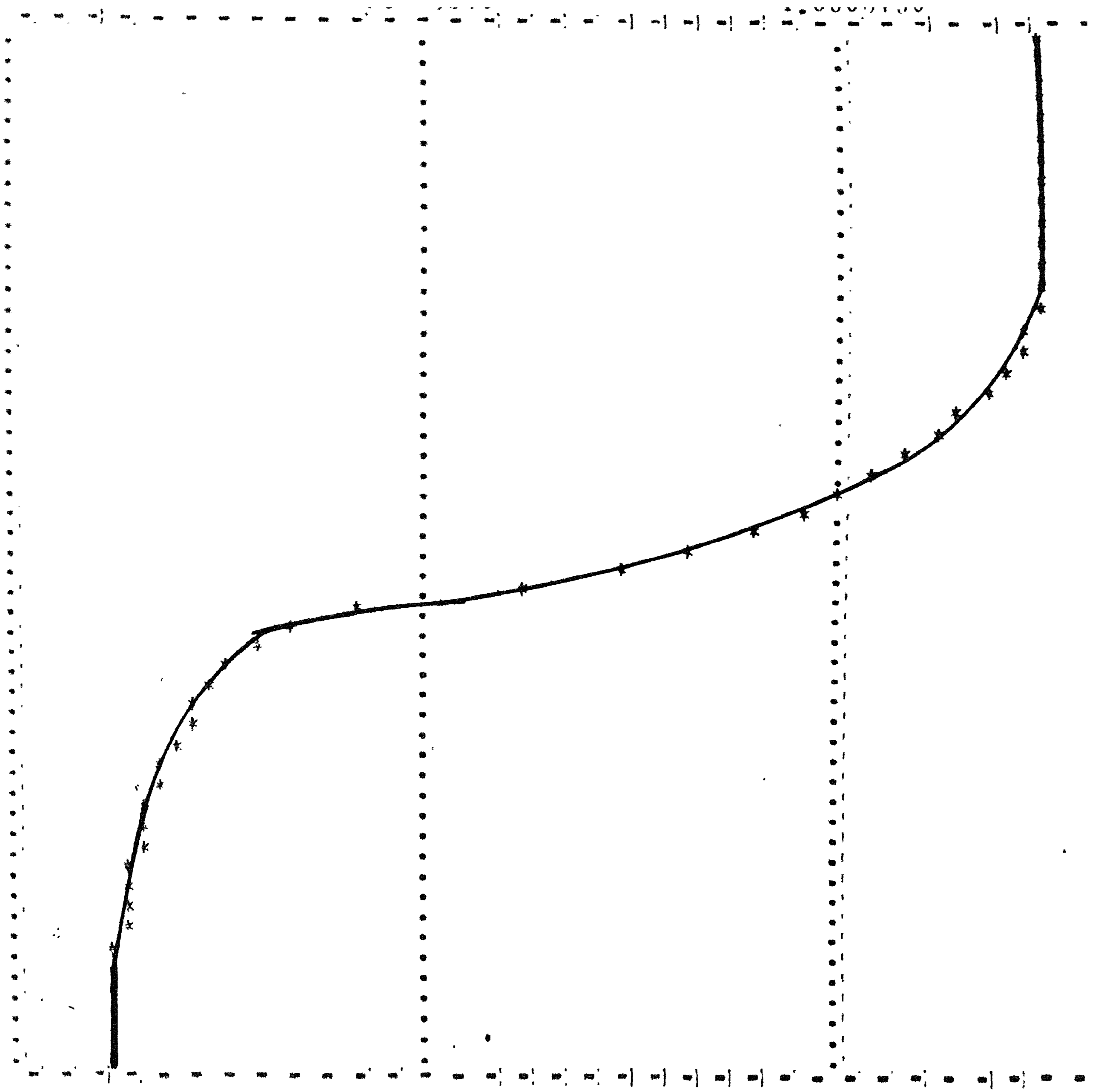
The DC transfer curve is shown in Fig. 4.1.

2. Transient response of an inverter (β_R = 1:4)

Table 4.2 SPICE deck

```
C1     3    0    0.3pF
VDD    4    0    DC    5V
V1     1    0    PULSE   0V   5V   30NS   20NS   20NS   1.0US   2.0US
M3     4    3    3     0    MDP   L=20U   W=5U
M1     3    1    0     0    MEN   L=5U    W=5U
.MODEL  MEN   NMOS   KP=3.0 E-5   VTO=1V
.MODEL  MDP   NMOS   KP=2.5 E-5   VTO=-4V
.TRAN   50NS   2US
.PLOT   TRAN   V(3)
.END
```

0.00D+00	5.[illegible]
1.00D-01	5.[illegible]
2.00D-01	5.[illegible]
3.00D-01	5.[illegible]
4.00D-01	5.[illegible]
5.00D-01	5.[illegible]
6.00D-01	5.[illegible]
7.00D-01	5.[illegible]
8.00D-01	5.[illegible]
9.00D-01	5.[illegible]
1.00D+00	5.[illegible]
1.10D+00	4.9[illegible]
1.20D+00	4.9[illegible]
1.30D+00	4.9[illegible]
1.40D+00	4.9[illegible]
1.50D+00	4.9[illegible]
1.60D+00	4.[illegible]
1.70D+00	4.[illegible]
1.80D+00	4.[illegible]
1.90D+00	4.[illegible]
2.00D+00	4.[illegible]
2.10D+00	4.[illegible]
2.20D+00	4.[illegible]
2.30D+00	3.[illegible]
2.40D+00	3.[illegible]
2.50D+00	3.[illegible]
2.60D+00	2.[illegible]
2.70D+00	2.[illegible]
2.80D+00	1.[illegible]
2.90D+00	1.[illegible]
3.00D+00	1.[illegible]
3.10D+00	1.[illegible]
3.20D+00	9.[illegible]D-01
3.30D+00	9.[illegible]D-01
3.40D+00	8.[illegible]D-01
3.50D+00	7.[illegible]D-01
3.60D+00	7.[illegible]D-01
3.70D+00	7.[illegible]D-01
3.80D+00	6.77[illegible]D-01
3.90D+00	6.[illegible]D-01
4.00D+00	6.[illegible]D-01
4.10D+00	5.[illegible]D-01
4.20D+00	5.71[illegible]D-01
4.30D+00	5.5[illegible]D-01
4.40D+00	5.31[illegible]D-01
4.50D+00	5.[illegible]D-01
4.60D+00	4.[illegible]D-01
4.70D+00	4.[illegible]D-01
4.80D+00	4.57[illegible]D-01
4.90D+00	4.537D-01
5.00D+00	4.[illegible]D-01



Y

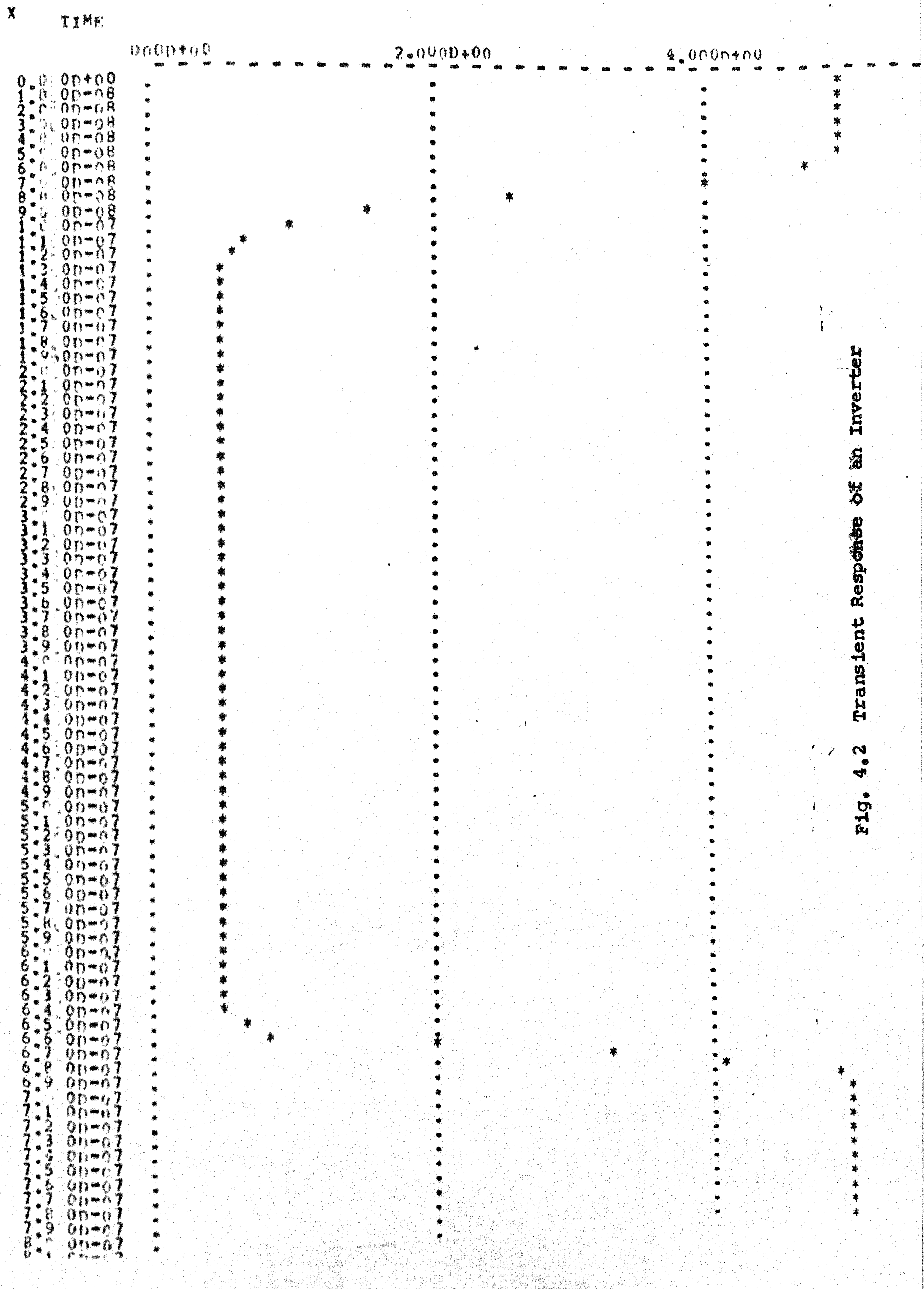


Fig. 4.2 Transient Response of an Inverter

The plot of the transient response is shown in Fig. 4.2

3. DC transfer curve of an inverter-negative going pulse (β_R=1:4).

Table 4.3 SPICE deck

```
C1   3    0  0.3pF
VDD  4    0  DC   5V
V1   1    0  PULSE  5V   0V   30NS  20NS  20NS  1.0US  2.0US
M3   4    3   3  0  MDP  L=20U  W=5U
M1   3    1   0  0  MEN  L=5U   W=5U
.MODEL   MEN   NMOS   KP=3.0 E-5  VTO=1V
.MODEL   MDP   NMOS   KP=2.5 E-5  VTO=-4V
.DC      D1    5V  0V  0.1
.PLOT    DC    V(3)
.END
```

The DC transfer curve is shown in Fig. 4.3.

4. Transient response of a 2-input NOR gate.

Table 4.4 SPICE deck

```
C1   3  0   0.3pF
VDD  4  0   DC  5V
V1   1  0   PULSE  0V  5V  30NS  20NS  20NS  1.0US  2.0US
V2   2  0   PULSE  0V  5V  30NS  20NS  20NS  1US    2US
M3   4  3   3   0   MDP  L=20U   W=5U
M1   3  1   0   0   MEN  L=5U    W=5U
M2   3  2   0   0   MEN  L=5U    W=5U
.MODEL  MEN  NMOS  KP=3.0 E-5  VTO=1V
.MODEL  MDP  NMOS  KP=2.5 E-5  VTO=-4V
.TRAN   50NS  2US
.PLOT   TRAN  V(3)
.END
```

The transient response plot is shown in Fig. 4.4.

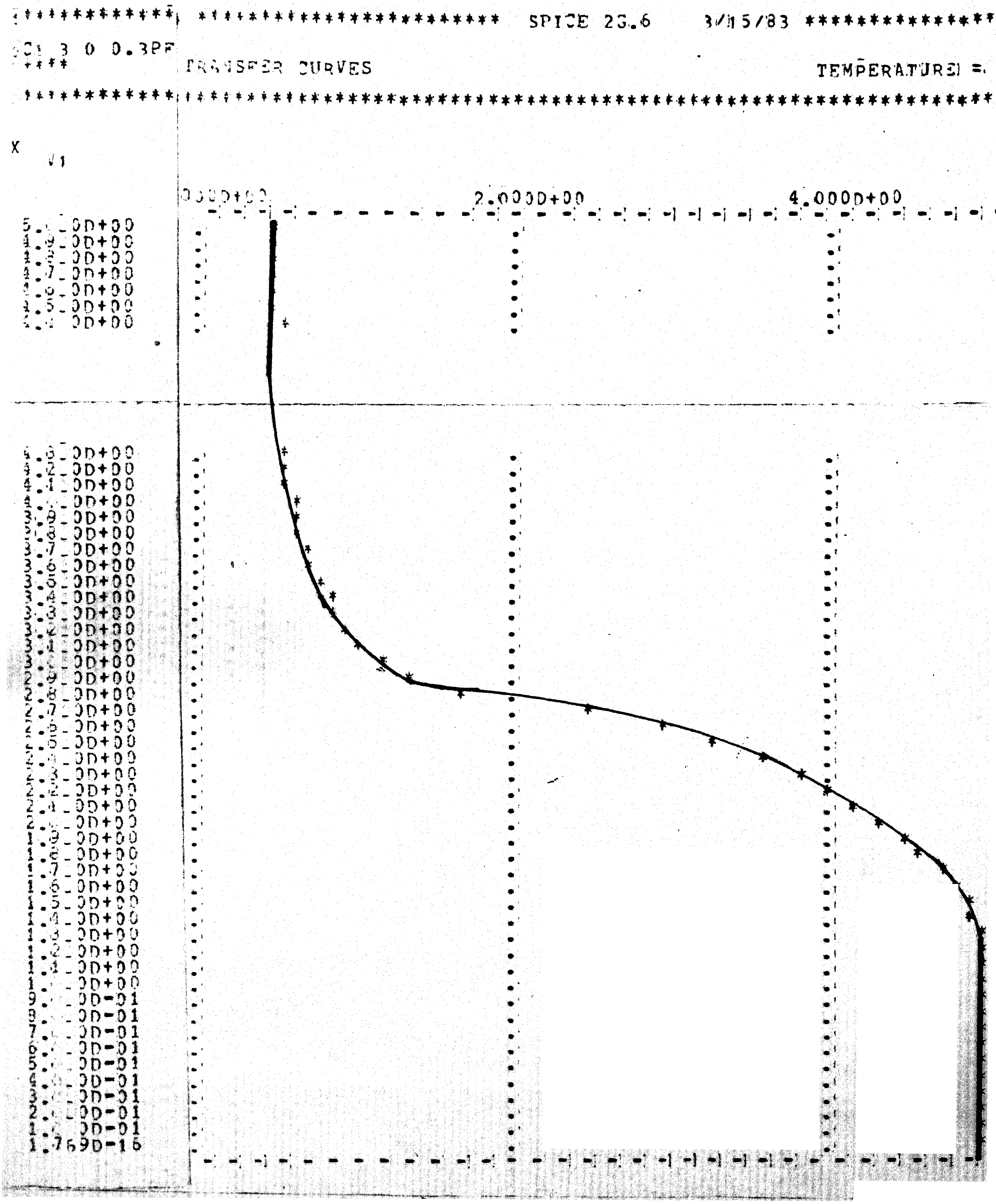


Fig. 4.3 DC Transfer Curve of an Inverter - negative going Pulse

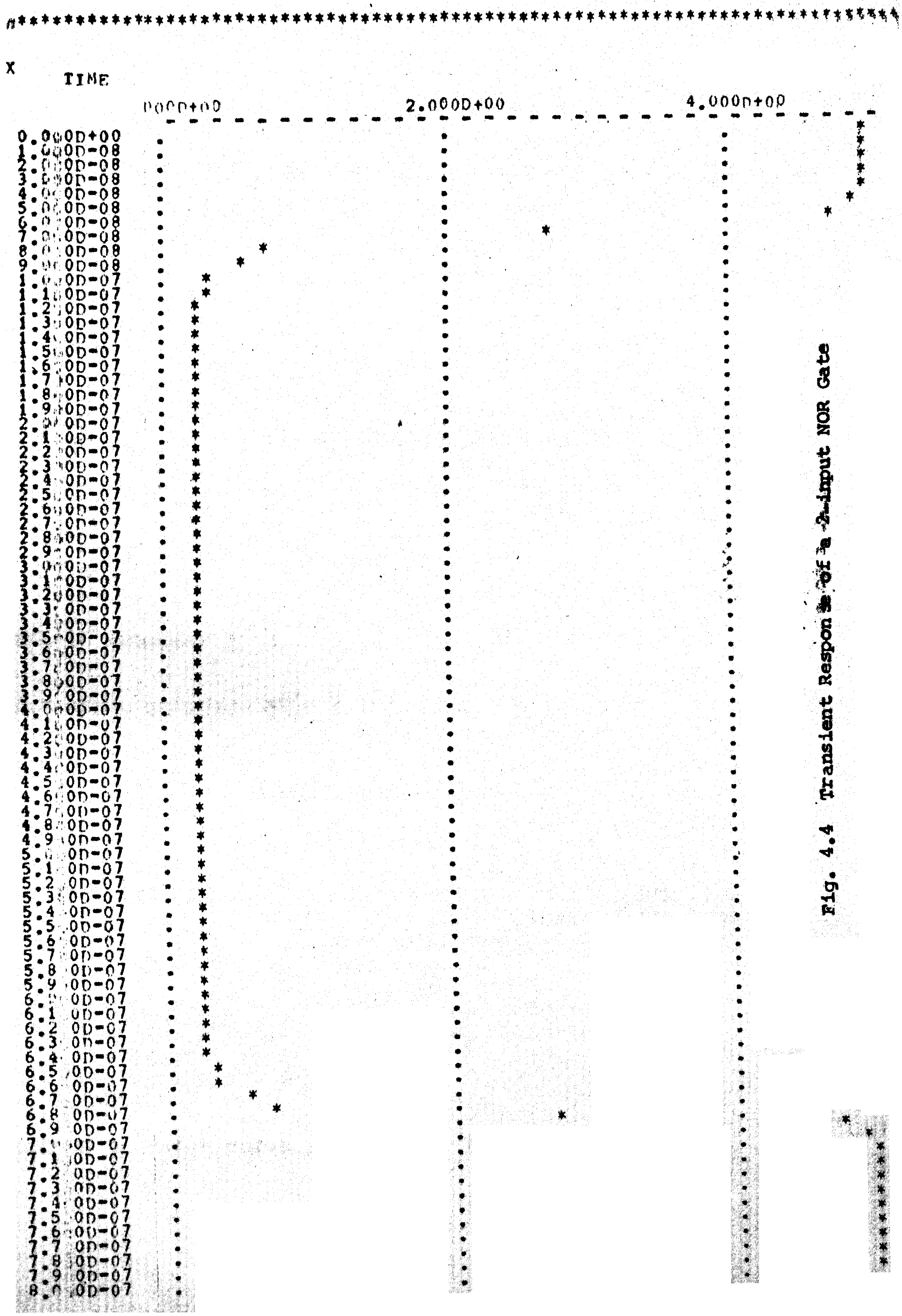


Fig. 4.4 Transient Response of a 2-input NOR Gate

5. DC transfer characteristics of a 2-input NAND gate-positve going input .

Table 4.5 SPICE deck

```
C1     4    0    0.3pF
VDD    5    0    DC      5V
V1     1    0    PULSE   5V    0V    20NS   20NS   20NS   3US   6US
V2     2    0    DC      5V
M1     3    1    0    0    MEN    L=5U    W=10U
M2     4    2    3    0    MEN    L=5U    W=10U
M3     5    4    4    0    MDP    L=20U   W=5U
.MODEL    MEN    NMOS    KP=3.0 E-5    VTO=1V
.MODEL    MDP    NMOS    KP=2.5 E-5    VTO=-4V
.DC       V1     5V      0V     0      2V
.PLOT     DC     V(4)
.END
```

The DC transfer curve is shown in Fig. 4.5.

6. Transient response of a 2-input NAND gate-negative going inputs.

Table 4.6 SPICE deck

```
C1     4    0    0.3pF
VDD    5    0    DC      5V
V1     1    0    PULSE   5V    0V    20NS   20NS   20NS   3US   6US
V2     2    0    PULSE   5V    0V    20NS   20NS   20NS   3US   6US
M1     3    1    0    0    MEN    L=5U    W=10U
M2     4    2    3    0    MEN    L=5U    W=10U
M3     5    4    4    0    MDP    L=20U   W=5U
.MODEL    MEN    NMOS    KP=3.0 E-5    VTO=1V
.MODEL    MDP    NMOS    KP=2.5 E-5    VTO=-4V
.TRAN     50NS   6US
.PLOT     TRAN   V(4)
.END
```

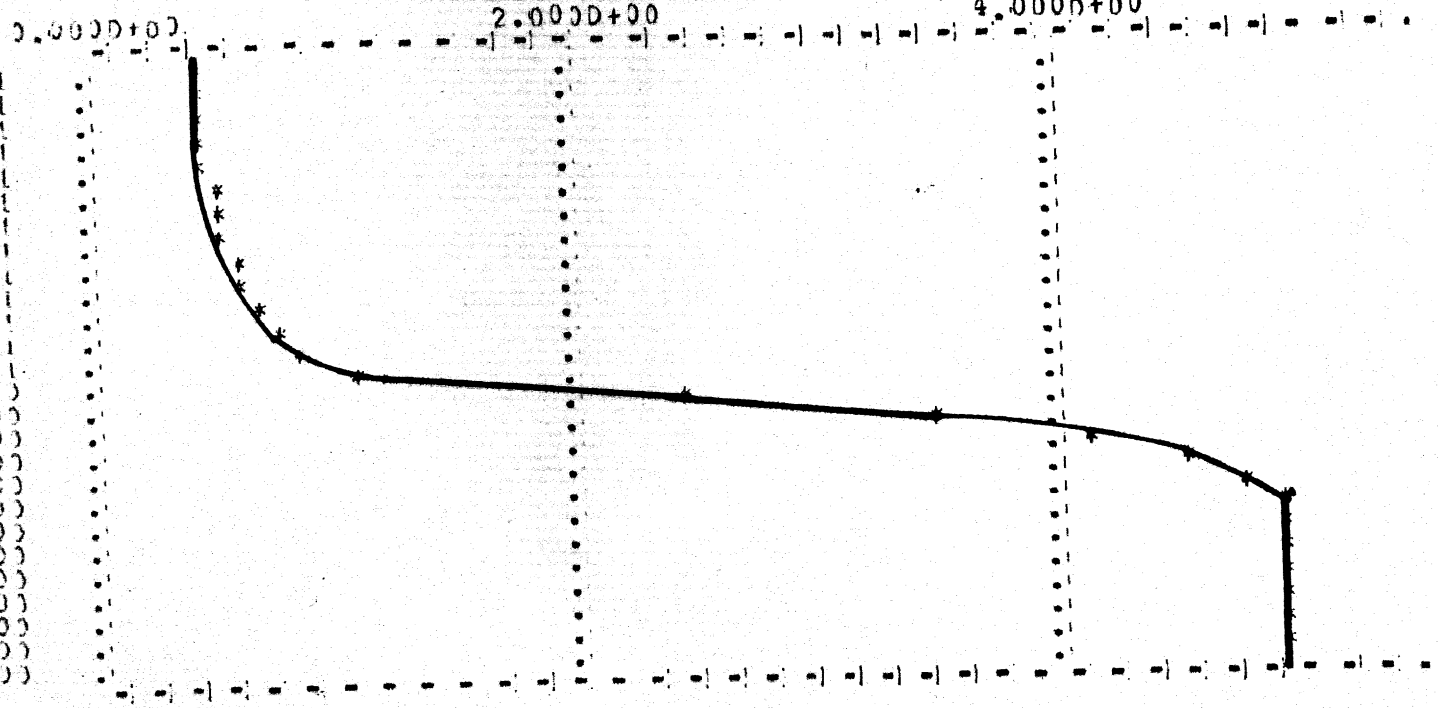
The DC transfer curve is shown in Fig. 4.6.

1*****************27 Dec 3 ************************* SPICE 2G.6 3/15/83 *********************

CC1 1 0 0.3PF DC TRANSFER CURVES TEMPERATURE = 27.0
****

****

X V1	V(4)
5.000D+00	4.410D-01
4.800D+00	4.537D-01
4.600D+00	4.680D-01
4.400D+00	4.842D-01
4.200D+00	5.025D-01
4.000D+00	5.240D-01
3.800D+00	5.489D-01
3.600D+00	5.785D-01
3.400D+00	6.145D-01
3.200D+00	6.594D-01
3.000D+00	7.171D-01
2.800D+00	7.953D-01
2.600D+00	9.432D-01
2.400D+00	1.134D+00
2.200D+00	2.477D+00
2.000D+00	3.530D+00
1.800D+00	4.439D+00
1.600D+00	4.642D+00
1.400D+00	4.803D+00
1.200D+00	4.952D+00
1.000D+00	5.000D+00
8.000D-01	5.000D+00
6.000D-01	5.000D+00
4.000D-01	5.000D+00
2.000D-01	5.000D+00
1.769D-16	5.000D+00



Y

Fig. 4.5 DC Transfer Curve of a 2-input NAND Gate-positve Going Pulses

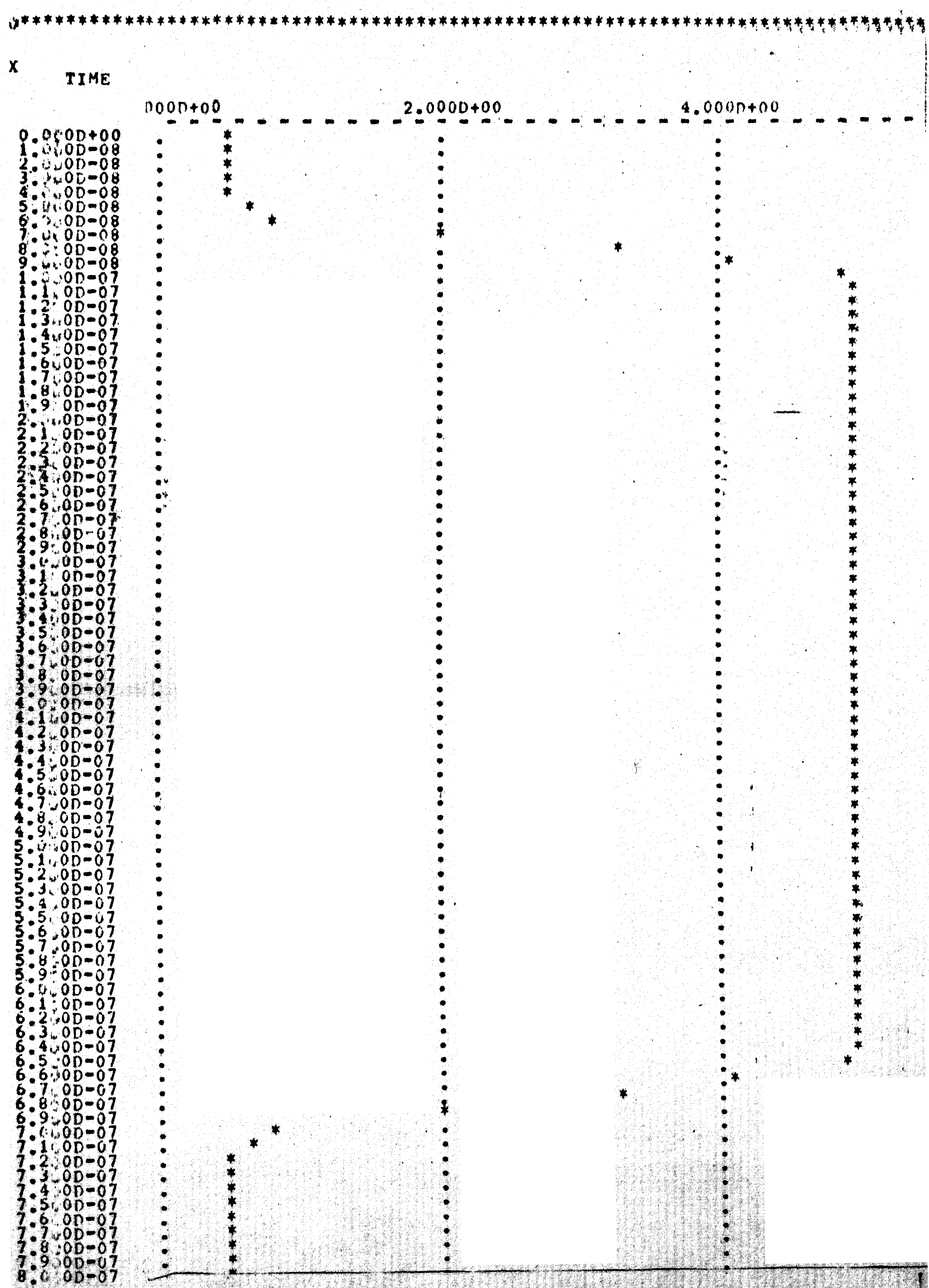


Fig. 4.6 Transient Response of a NAND Gate-negative Going Pulses

7. Transient response of a 2-input NAND gate-positive going inputs.

Table 4.7 SPICE deck

```
C1     4    0    0.3pF
VDD    5    0    DC      5V
V1     1    0    PULSE   0V    5V    20NS    20NS    20NS    3US    6U
V2     2    0    PULSE   0V    5V    20NS    20NS    20NS    3US    6US
M1     3    1    0       0     MEN   L=5U    W=10U
M2     4    2    3       0     MEN   L=5U    W=10U
M3     5    4    4       0     MDP   L=20U   W=5U
.MODEL  MEN   NMOS   KP=3.0   E-5   VTO=1V
.MODEL  MDP   NMOS   KP=2.5   E-5   VTO=-4V
.TRAN   50NS  6US
.PLOT   TRAN  V(4)
.END
```

The transient response plot is shown in Fig. 4.7.

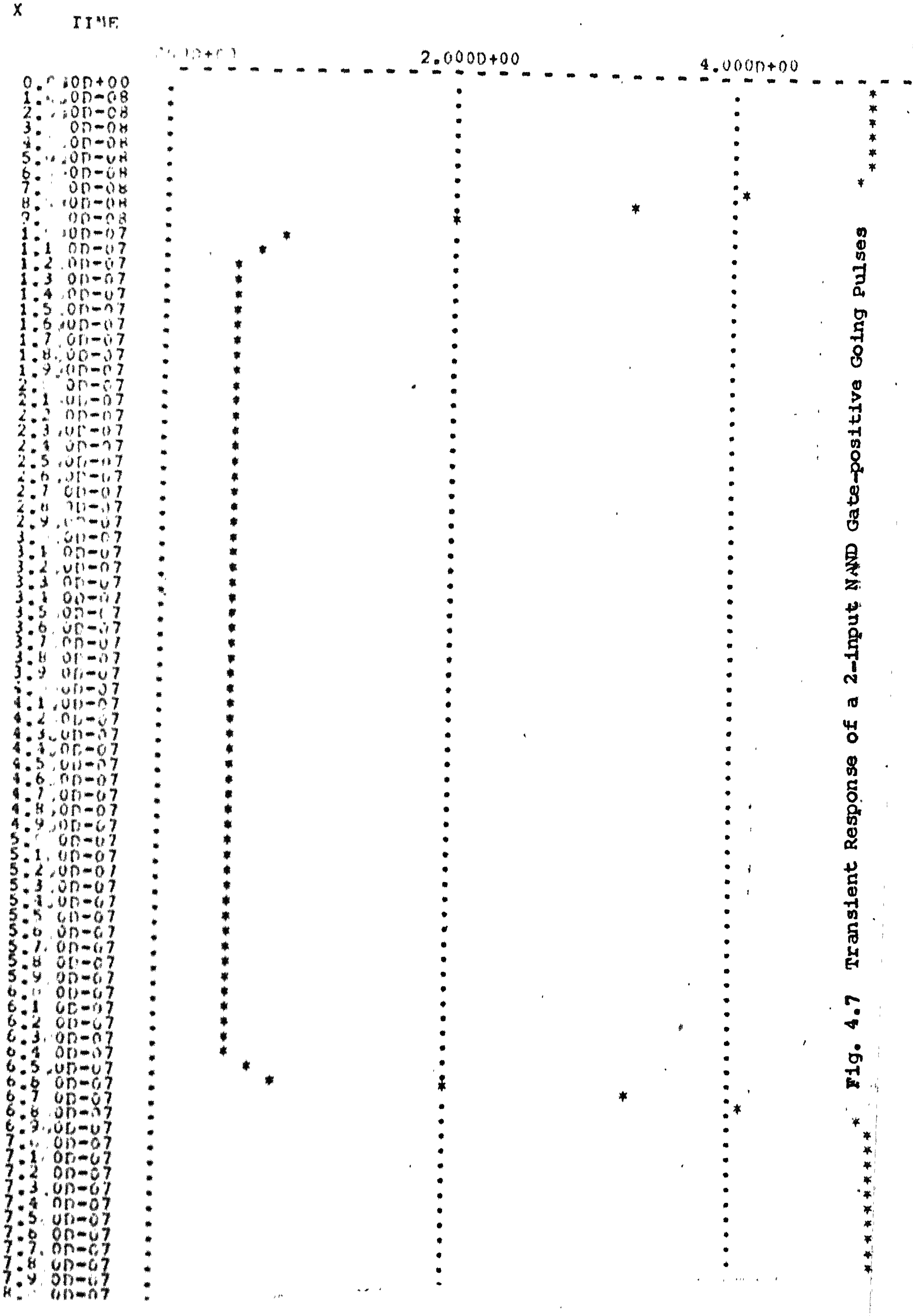


Fig. 4.7 Transient Response of a 2-input NAND Gate-positive Going Pulses

CHAPTER 5

RESULTS AND CONCLUSIONS

5.1 Stick diagram and Layout of some Functional Modules

The three software packages discussed in the Chapter 3, were used to generate the stick diagrams and the layouts of the functional modules given below.

1. Half Adder

The logic functions for the outputs, S (sum) and carry (C) with inputs A and B of this module are,

$$S = AB' + A'B,$$

$$C = AB.$$

The file 'INPUT' for Half adder is shown in Table 5.1.

Table 5.1

INPUT file for the Half-Adder

```
2   3   2   2.0   0.0   3.0
AB
AB'
A'B
AB
1   2
3
```

The stick diagram and the layouts are shown in Figs. 2.12 and Fig. 5.1 respectively.

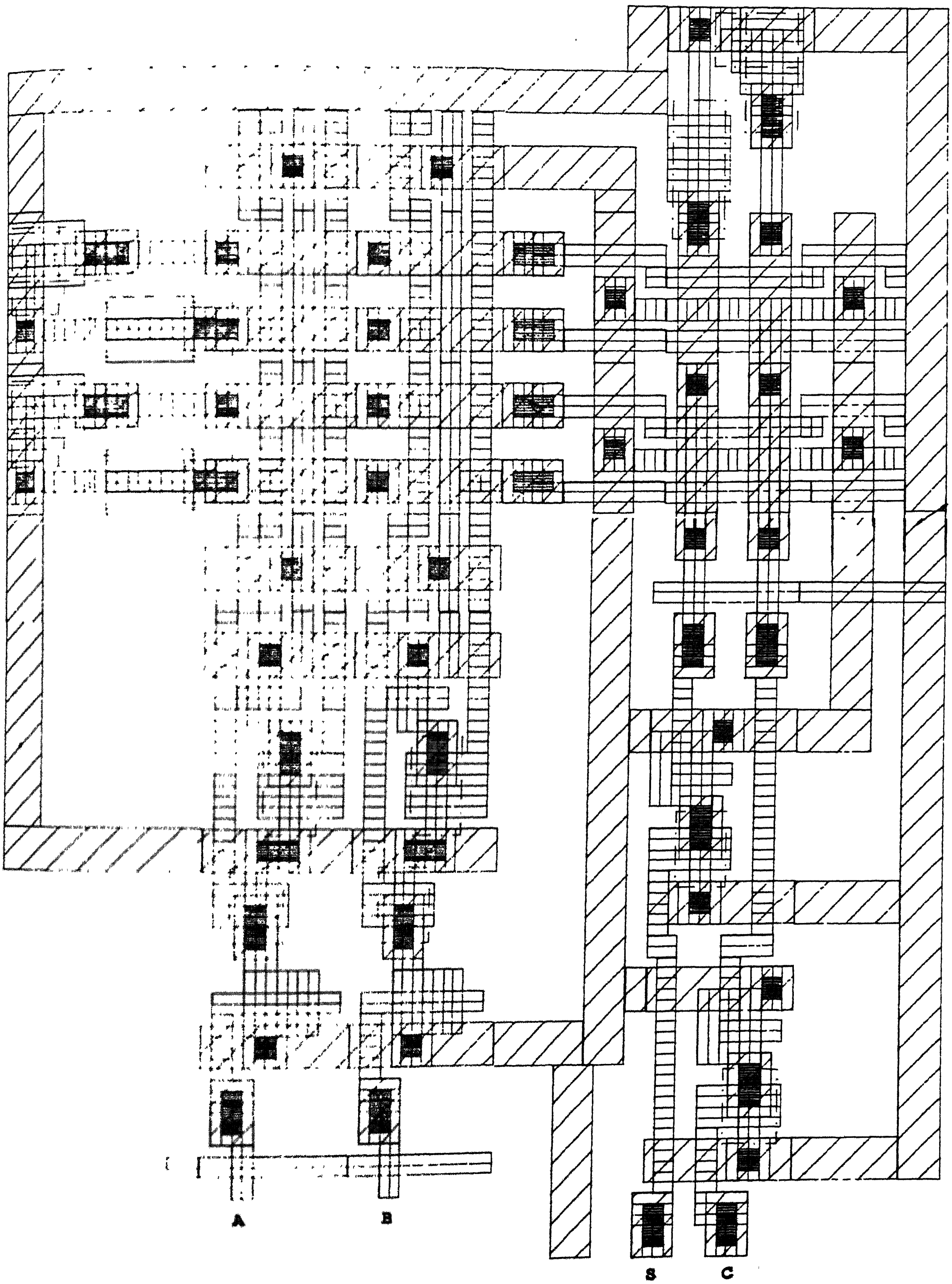


Fig. 5.1 Layout of the Half-Adder

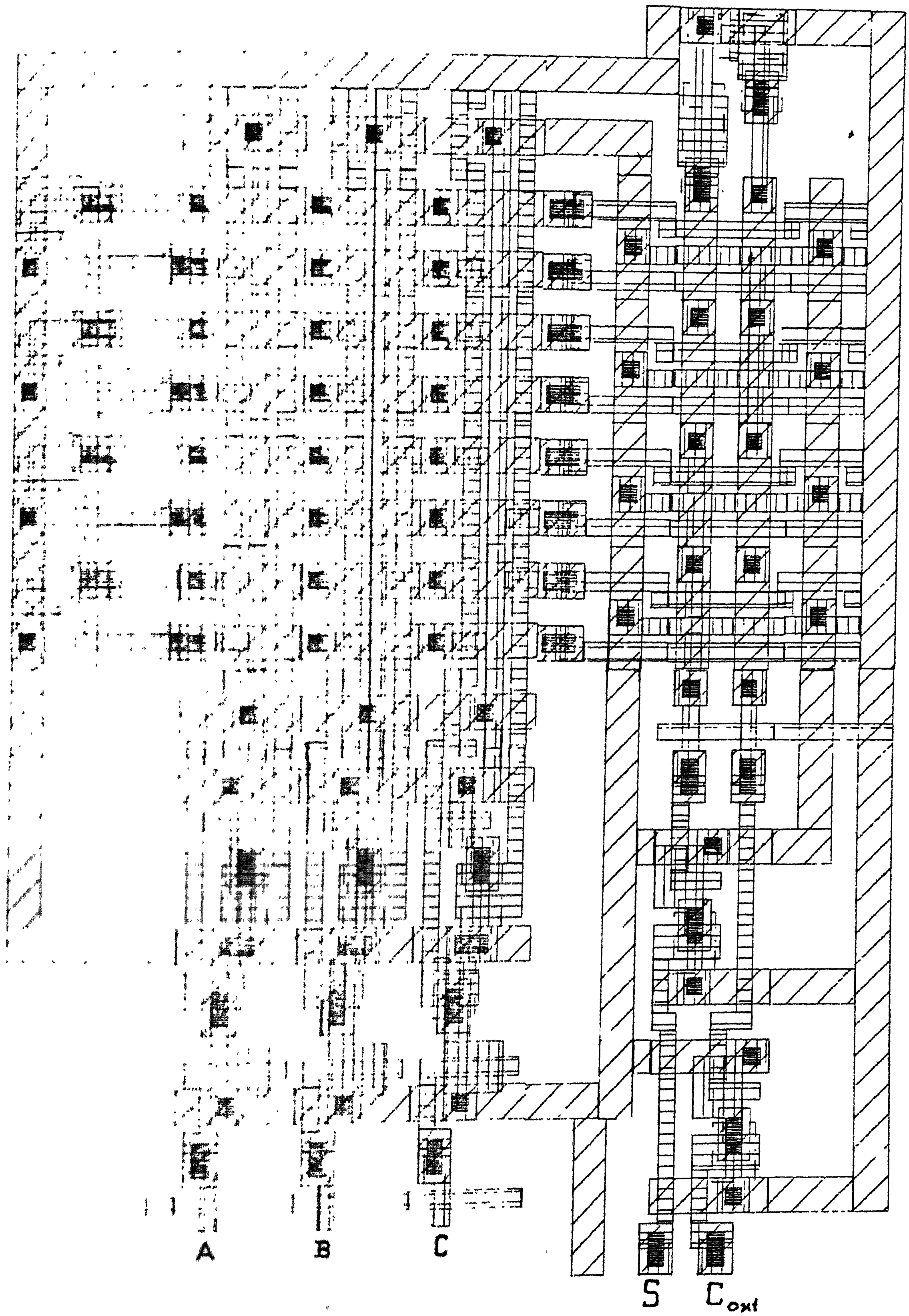


Fig. 5.2 Layout of the Full-Adder

2. <u>Full Adder (one bit)</u>

The outputs sum (S) and carry (C_{out}), of the full adder with addend bit A, augend bit B, and carry-in (C) are given by the logic functions,

$$S = ABC + AB'C' + A'BC' + A'B'C,$$

$$C_{out} = AB + BC + AC.$$

For the file INPUT, see Table 3.1 of Chapter 3. for layout see Fig.5.2

3. <u>Binary to Gray Code Code Converter (4 bits)</u>

4 bit Binary Code				4 bit Gray Code			
A	B	C	D	G_4	G_3	G_2	G_1
(MSB)			(LSB)	(MSB)			(LSB)

<u>Logic functions :</u>

$G_1 = CD' + C'D.$

$G_2 = BC' + B'C.$

$G_3 = AB' + A'B.$

$G_4 = A.$

The file INPUT for functional module is shown in table 5.2.

Table 5.2

<u>INPUT file for Binary to Gray Code Converter</u>

```
4     7     4     0.0     0.0     4.0
ABCD
A
AB'
A'B
BC'
B'C
CD'
C'D
1
2     3
4     5
6     7
```

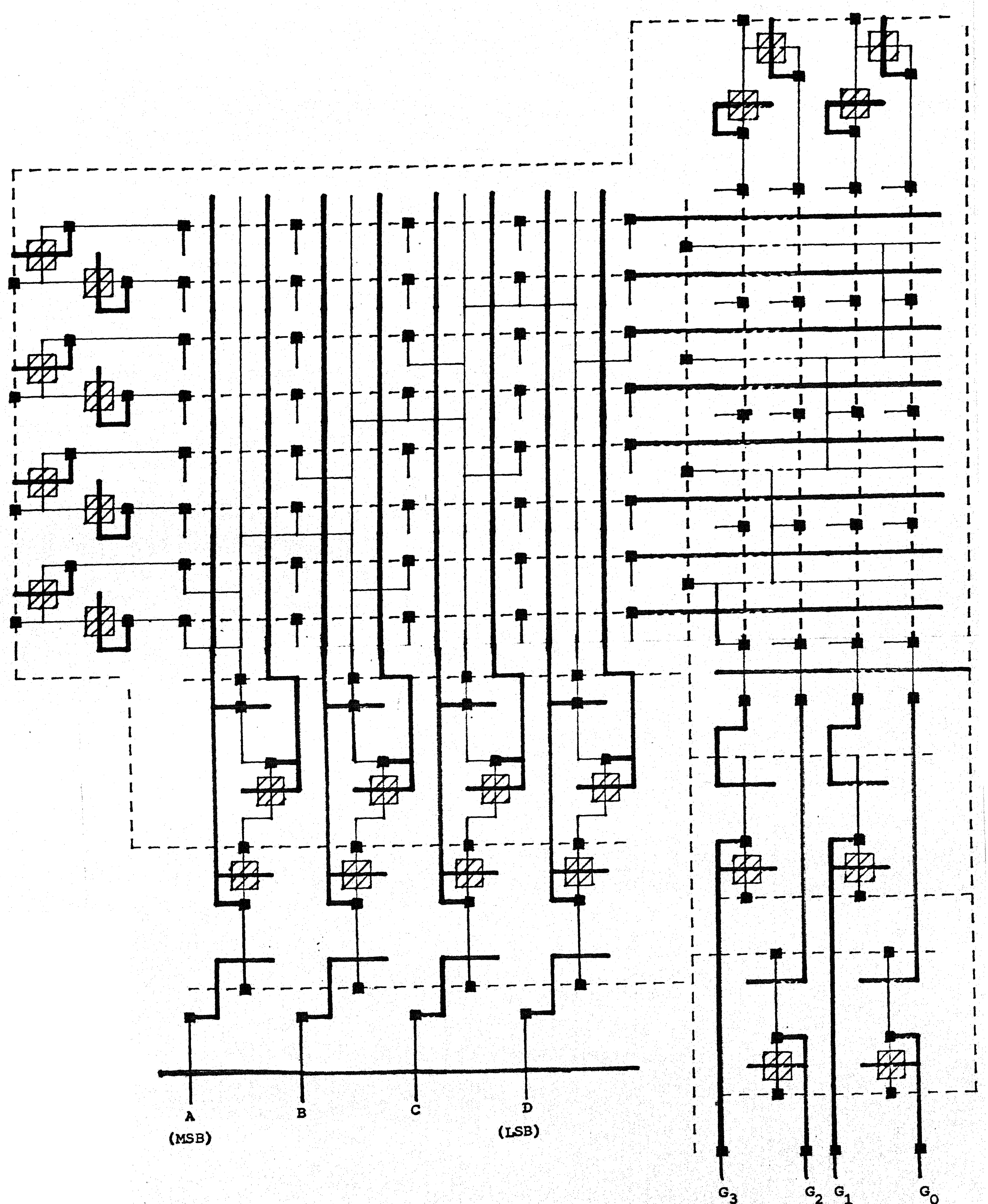


Fig. 5.3 Stick Diagram of the Binary to Gray Code Converter

The stick diagram and the layouts are shown in Fig. 5.3 and 5.4 respectively.

4. Decade Counter

Z	Y	X	W	W_{n+1}	X_{n+1}	Y_{n+1}	Z_{n+1}
(LSB)			(MSB)	(MSB)			(MSB)

The logic functions are given by ,

$$W^{n+1} = XYZ + WZ',$$
$$X^{n+1} = XY' + XZ' + X'YZ,$$
$$Y^{n+1} = W'Y'Z + YZ' \quad \text{and}$$
$$Z^{n+1} = Z'$$

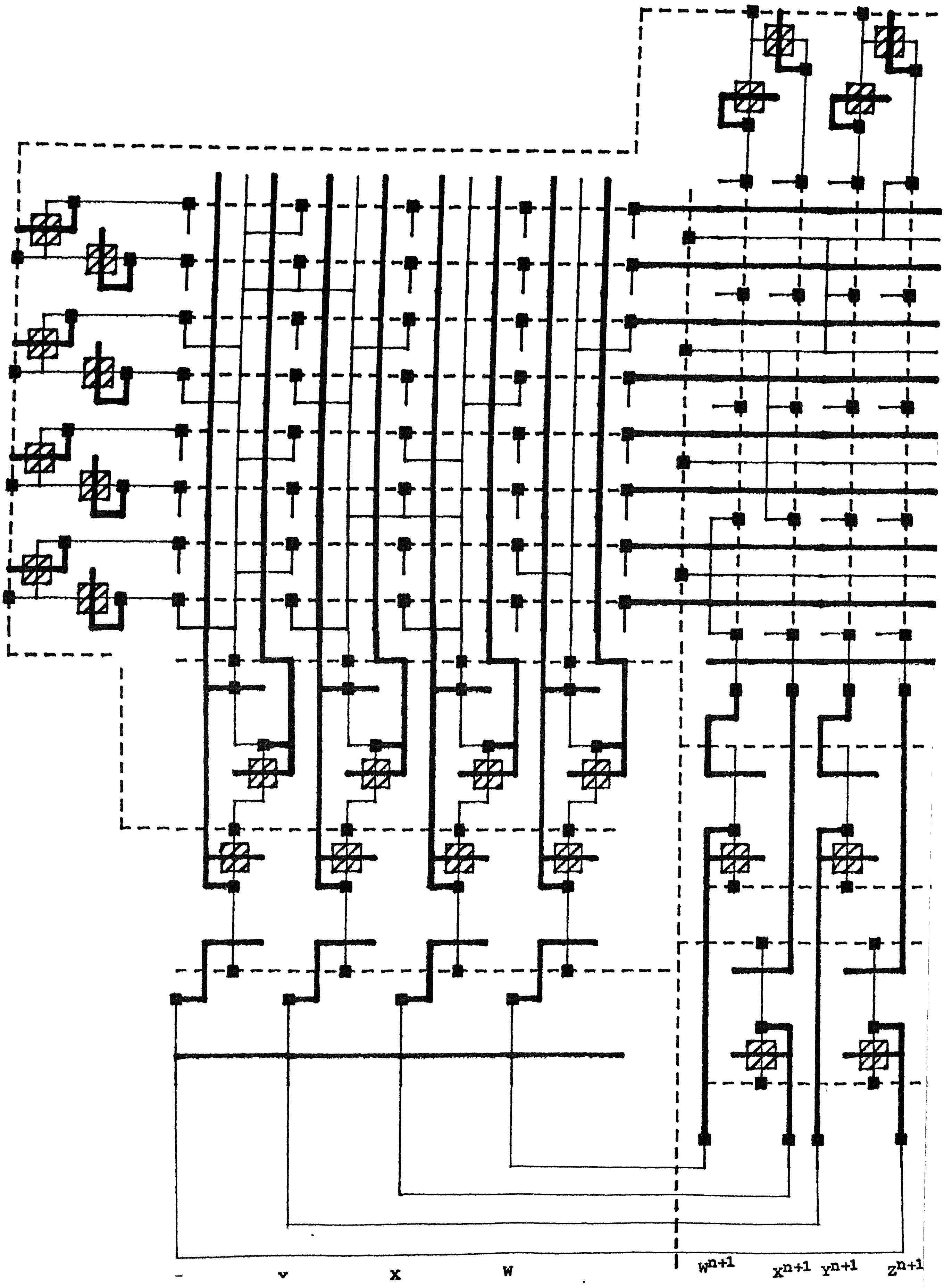
The file INPUT is given in Table 5.3.

Table 5.3

INPUT file for Decade Counter

```
4    8    4   0.0   0.0   3.0
ZYXW
XYZ
WZ'
XY'
XZ'
X'YZ
W'Y'Z
YZ'
Z'
1    2
3    4    5
6    7
8
```

ıe stick diagram is shown in Fig. 5.5.



v
x
w
W^{n+1}
X^{n+1}
Y^{n+1}
Z^{n+1}

5. 4:1 Digital Multiplexer

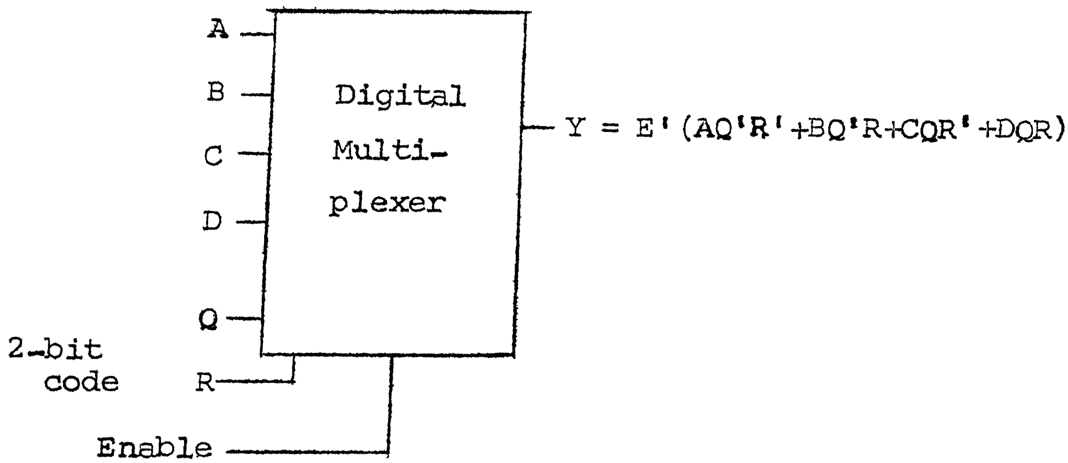


Fig.5.6 Block schematic of 4:1 digital multiplexer.

The file INPUT is shown in table 5.4.

Table 5.4

INPUT file for Multiplexer

```
7  4  1  0.0  0.0  3.0
ABCDQRE
AQ'R'E'
BQ'RE'
CQR'E'
DQRE'
1  2  3  4.
```

The stick diagram and the layout are shown in fig. 5.7 and Fig. 5.8 respectively.

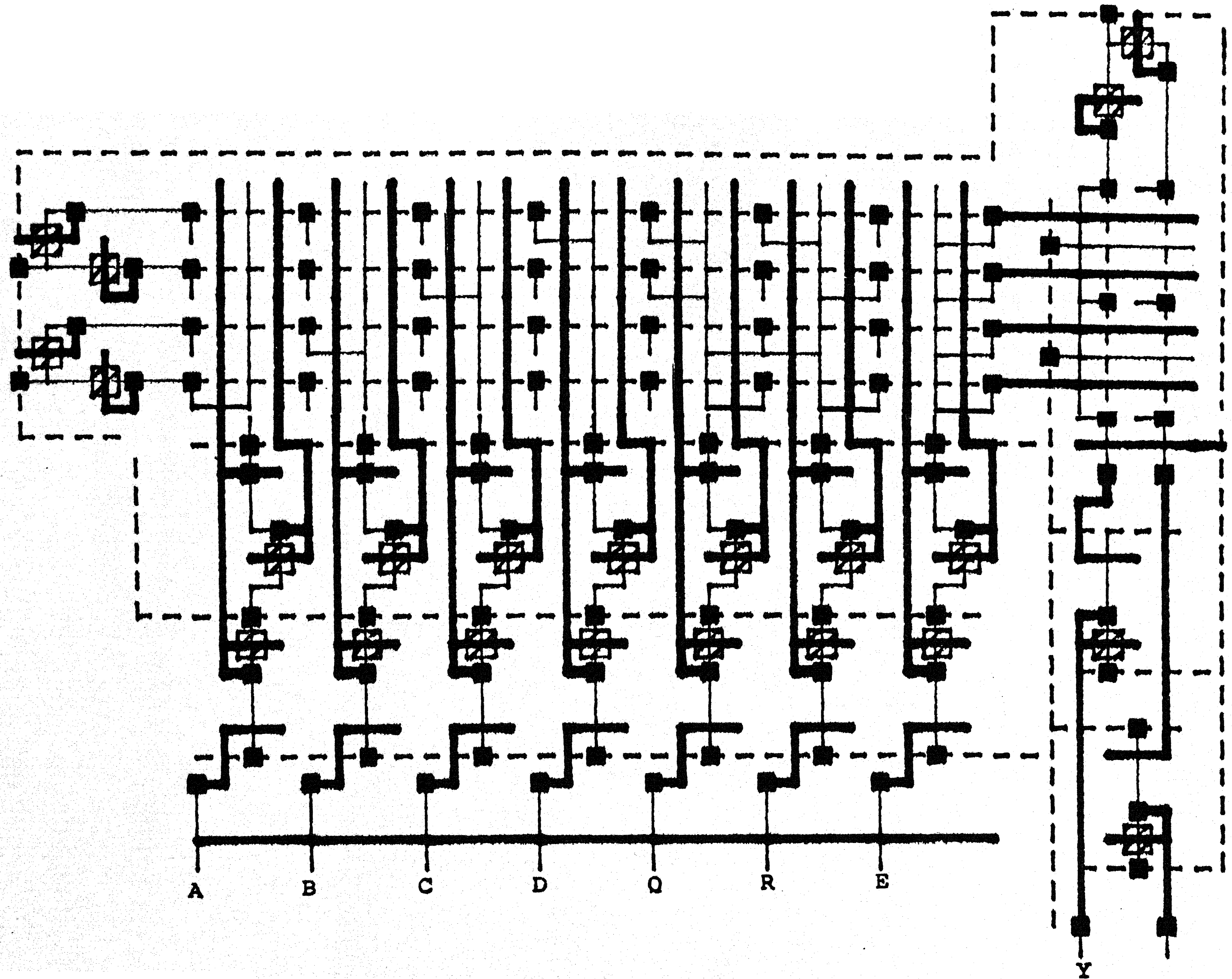


Fig. 5.7 Stick Diagram of 4:1 Digital Multiplexer

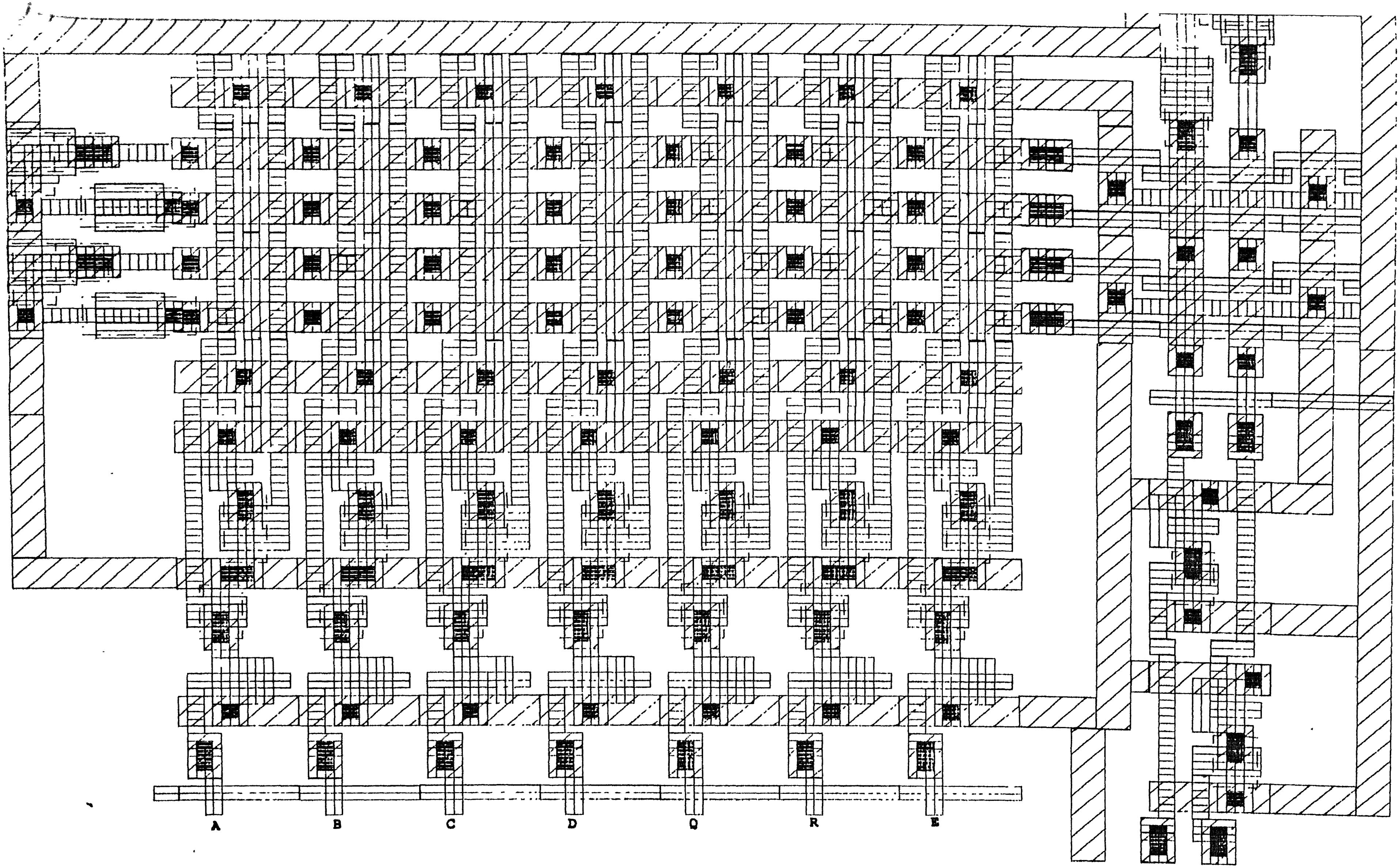


Fig. 5.8 Layout of 4:1 Digital Multiplexer

6. <u>1:4 De-multiplexer, 2:4 Decoder.</u>

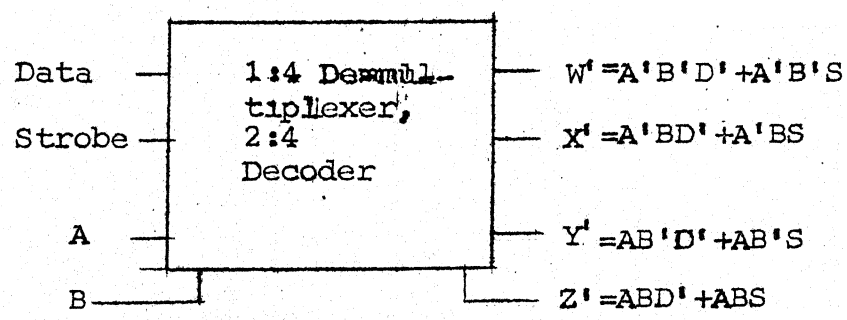


Fig. 5.9 1:4 De-multiplexer, 2:4 Decoder. The INPUT file is shown in Table 5.5.

Table 5.5

INPUT FILE for De-multiplexer

```
4  8   4  0.0   0.0  1.0

ABDS

A'B'D'

A'B'S

A'BD'

A'BS
1  2  3  4
AB'D'

AB'S

ABD'

ABS

1   2

3   4

5   6

7   8
```

The stick diagram of the demultiplexer is shown in Fig. 5.10.

Note : The realization gives inverted outputs. Use of external inverters is recommended to get W,X,Y and Z.

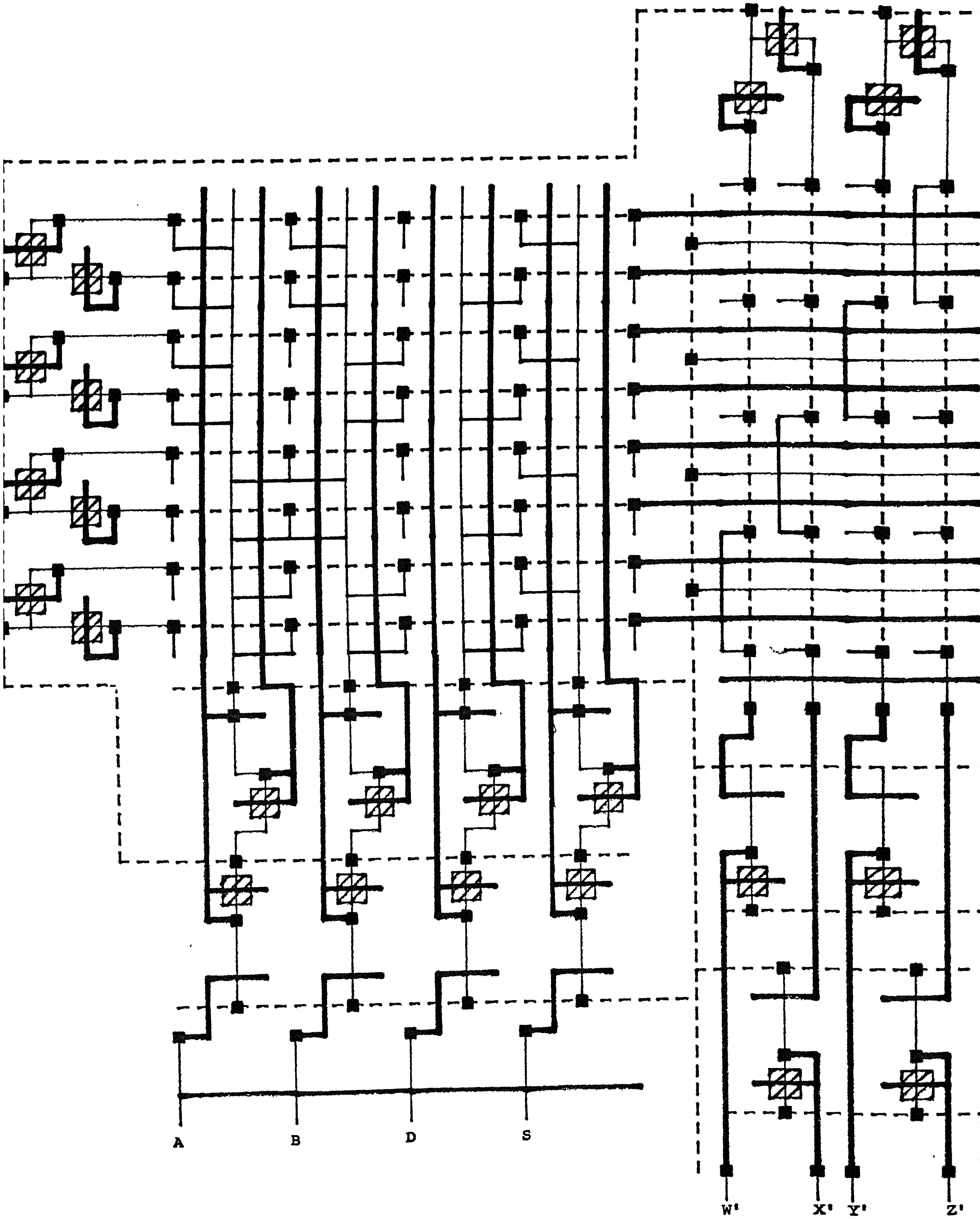


Fig. 5.10 Stick Diagram of the De-multiplexer

5.2 Conclusions

In this work design of LSI/VLSI functional cells has been discussed with emphasis on structured design approach (PLA). Computer-aids for the generation of stick-diagram and layout of PLA macros have been developed. The stick diagram and layout of many functional modules have been generated using the programs developed. The functional cells have been simulated for their functional behaviour and electrical characteristics. While PLA macros have already appeared in commercially available microprocessors for use in control logic, it is now becoming practical to use them for the Arithmetic Logic Unit as well. This will permit the use of common automated design processes for both control logic and data paths.

If logic minimization is included in the software packages for the generation of stick diagram and layout, the packages will become more elegant.

REFERENCES

1. C.A. Mead and L.A. Conway, 'Introduction to VLSI Systems', Addison-Wesley, Reading, Mass., 1980.
2. J. Mavor, M.A. Jack, P.B. Denyer, 'Introduction to MOS LSI Design', Addison-Weslay publishing company, 1983.
3. W.N. Carr and J.P. Mize, 'MOS/LSI Design and Applications', McGraw-Hill Book Co., Inc., New-York, 1972.
4. L.O. Chua, P.M. Lin, 'Computer-Aided Analysis of Electronic Circuits', Prentice-Hall Inc., Eaglewood Cliffs, New Jersey 1975.
5. E.V. Krishnamurthy and S.K. Sen, 'Computer-Based Numerical Algorithms', East-West Press, 1976.
6. A.R. Newton, D.O. Pederson, A. Sangiovanni-Yincentelli, 'SPICE Version 2G User's Guide', University of California, Burkeley, C.A. 1981.
7. V. Rajaraman, 'Computer Programming in Fortran IV', 2nd Ed. Prentice Hall of India, 1982.
8. Computer Programming in PASCAL, 'V. Rajaraman', Prentice Hall of India, 1983.
9. David F. Rogers and J. Alan Adams, 'Mathematical Elements for Computer Graphics', McGraw-Hill Book Company, 1979.
10. Ian O. Angell, 'A Practical Introduction to Computer Graphics' Macmillan Computer Science Series, 1981.
11. D.E. Knuth, 'The Art of Computer Programming', Volume 1/ Fundamental Algorithms, Second Ed., Reading, Mass., Addison-Wesley Publishing Company, 1982.

12. Kathleen Jensen, Niklaks Wirth, 'User Manual and Report', Second Ed., Narosa Publishing House, New Delhi, 1983.

13. The PLOT-10, IGL User Manual.

14. C.M. Lin, 'PLA-Based Macros', JSSC, SC-16, April 1982, pp. 103-107.

15. Beke, W.M.C. Sansen, and R. Van Overstraeten, 'CALMOS : A Computer-Aided Layout Program for MOS/LSI', JSSC, June 1977, pp 281.

16. Jefferey H. Hoel, 'Some Variations of Lee's Algorithm', IEEE Trans. on Computers, Vol. C-25, No.1, January 1976, pp 19-24.

17. David K. Lynn, 'Computer-Aided Layout System For Integrated Circuit', IEEE Trans. on Circuit Theory, January 1971, pp 128-139.

18. Yahiko Kambayashi, 'Logic Design of Programmable Logic Arrays', IEEE Trans. on Computers, Vol. C-28, No. 9, Sept. 1979, pp. 609-616.

19. Martin S. Schmookler, 'Design of Large ALUS Using Multiple PLA Macros', IBM Journal of R and D, Vol. 24, No.1, January 1980, pp 2-14.

20. James M. Geyer, 'Connection Routing Algorithm for Printed Circuit Boards', IEEE Trans. on Circuit Theory, Jan. 1971, pp. 95-100.

21. Weinberger, 'High Speed PLA Adders', IBM Journal of R and D, 1979, vol. 23, pp. 163-178.

22. Special Issue on VLSI, IEEE Trans. on Electron Devices, April 1979, pp. 341.

APPENDIX A

USER'S GUIDE

'Computer aids for Random Logic Implementation','PLA Stick-diagram Generator', and 'PLA Layout Generator' are general purpose NMOS stick diagram and layout generator programs. The uses of these packages are explained below.

A.1 Computer-aids for Random Logic Implementation

This package is in the file named 'RANDOM' and has the following subroutines which are accessible to the user.

A.1.1 Basic building blocks BOX(ITYPE, XO, YO, XLENTH, YLENTH)

The subroutine 'BOX' draws a box with either of the following types depending on the value of 'ITYPE'.

(a) box with horizontal lines to represent poly region for ITYPE=1,

(b) box with vertical lines to represent diffusion region " =2,

(c) box with lines inclined at 45 degress to the horizontal to represent metalization for ITYPE = 3,

(d) box outlined by a chained line to represent Ion-implementation region for ITYPE = 4,

(e) completely blackened box to represent a contact cut for ITYPE = 5.

XO, YO : x and y co-ordinates of the lower left corner of the BOX.

xlenth : length of the box in mm.

ylenth : height of the box in mm.

INVERTER (XO, YO, DW, DL, LW, LL, LAM)

Draws the layout of an inverter.

Note that all the arguments are REAL variables.

XO, YO : x and y co-ordinates of the lower left corner of the inverter.

DW, DL : Drive channel width and length respectively

LW, LL : Load channel width and length respectively.

LAM : Minimum feature size factor (2xlam=minimum feature size).

Example A.1

```
CALL INVERTER (0.5, 5.0, 2.0, 1.0, 1.0, 4.0, 3.0)
STOP
END
```

The above program segment draws an inverter with (W/L) driver = 2.0/1.0 and (W/L) load = 1.0/4.0 with minimum line width = 6.0 mm at x = 0.5 mm and y = 5.0 mm with respect to the current origin. The file containing the above program segment is executed by the following monitor command.

```
.Ex FILENAME. EXT, RANDOM,/SEA SYS:IGL.REL
```

IGL.REL is the file containing the compiled code of Interactive Graphic Library subroutines. 'RANDOM' is the name of the file containing 'Computer-Aids for Random Logic Implementation'.

NOR (Inputs, xo, yo, DW, DL, LW, LL, LAM)

This subroutine draws the layout of a multi-input NOR gate.

Inputs : No. of inputs in the NOR gate.

xo, yo : X and Y co-ordinates of the lower left corner of the NOR gate layout.

DW, DL : Drive channel width and length respectively.

LW, LL : Load channel width and length respectively.

2XLam : Minimum feature size.

Example A.2 The program segment

```
CALL NOR(3,0.0, 0.0, 1.0, 1.0, 1.0, 4.0, 2.5)
STOP
END
```

draws the layout of a 3-input NOR gate with (W/L) driver = 1 and (W/L) load = 1/4 with the lower left corner at the current origin with 5 mm thick (minimum) lines.

NAND (INPUT, xo, yo, DW, DL, LW, LL, LAM)

All the arguments have the same meaning as in the NOR gate. This routine draws the layout of a multi-input NAND gate.

Example A.3

```
CALL NAND (2, 0.0, 0.0, 2.0, 1.0, 1.0, 4.0, 2.0)
STOP
END
```

The above program segment draws the layout of a two-input NAND

with (W/L) driver = 4 and (W/L) load = 1/4 so as to have the overall β_R =(1/No. of inputs)x(W/L) driver/(W/L) load) equal to 4. Note that the aspect ratio of the drivers are modified according to the number of inputs.

A.1.2 More Complex Structure – the Shift Register

Shift register is an example of a complex circuit, the layout of which can be generated by calling the inverter, the pass transistor and the necessary inter-connections repeatedly depending on the number of bits. This should make it very clear that the layout of any complex circuit can be very easily generated by making calls to layer description routine (BOX) and the necessary building blocks explained earlier.

General form : SREG (NOBITS, XO, YO, DW, DL, LW, LL, LAM)

NOBITS = No. of bits in the shift register.

All other arguments are similar to the NOR subroutine.

Example A.4

The program segment

```
Call SREG (4, 0.0, 0.0, 2.0, 1.0, 1.0, 4.0, 2.0)
Stop ; END
```

is executed to draw a 4-bit shift register with (W/L) driver = 2 (W/L) load = (1.0/4.0) and a pass transistor of unity aspect ratio at the current origin (x = 0.0, y = 0.0).

A.2 Stick Diagram Generation

All the routines needed to draw the stick diagram of a PLA macro are in the file named 'STK'.

The personality matrix for the PLA stick diagram and the layout is generated by a PASCAL program in a file named 'OUTPUT'. The input to the stick diagram generation program are the files 'INPUT' containing the logic function definition (to be described later) and 'OUTPUT'. The PASCAL and FORTRAN files are linked in a file named 'STICK.MIC'. Any file with an extension 'MIC' can contain monitor commands (in DEC 1090 System) and is executed by the monitor command '.DO file name'.

The contents of the file STICK.MIC are,

. Ex MAT.PAS

. CRLF (carriage return and linefeed)

. Ex STK,/SEA SYS:IGL.REL

'MAT.PAS' is the name of the PASCAL program file which produces the personality matrices.

A.2.1 Format of the file 'INPUT'

The name of the file containing the logic function definition, whose stick diagram is to be generated must be 'INPUT' and shall have the following information.

Table A.1

'INPUT' file for the Stick Diagram Generator

Line No.	Contents
1	No. of inputs, No. of product terms, No. of outputs, XO, YO, Lam (minimum feature size factor).
2	List of input variables.
3	1st product term (This will be left most output of the PLA).
<4>	<2^{nd} product term>.
. . .	
< n+2 >	<n^{th} product term >
(n+2)+1	1st output (expressed in terms of the product terms
. . .	
< n+2+m >	mth output

Total No. of lines = No. of product terms+No. of outputs+2. Consider the example of the Full-adder. It has 3 inputs A, B, and C and two outputs S and C_{out}, given by the Boolean expression,

$$S = ABC+AB'C'+A'BC'+A'B'C \text{ and}$$

$$C_{out} = AB + BC + AC$$

The entries 5.0, 0.0, and 3.0 in the first line of the table A.2 are for the X-coordinate, Y-coordinate, and the minimum feature size factor respectively. The lines for the input variables list and the product terms are free formated. This means that blanks are ignored, if present, between characters. The 10th line of the table, with entries 1,2,3 and 4 indicates that the first output, viz., the SUM is to be formed by ORing the first, second, third, and the fourth product terms namely, ABC, AB'C', A'BC' and A'B'C. The last line with entries 5,6 and 7 is to form the second output (the last in this case), by ORing the 5th, 6th and 7th product terms, i.e, AB, BC, and AC.

The stick diagram outline looks as shown in Fig. A.1.

The stick diagram (block diagram) is shown in Fig. A.2.

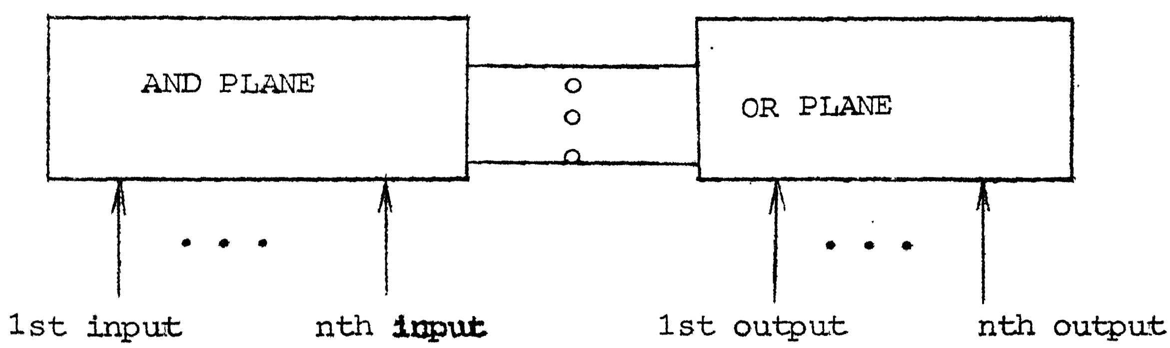


Fig. A.1 General block schematic of the Stick diagram

The product terms are,

(1) ABC

(2) AB'C'

(3) A'BC'

(4) A'B'C

(5) AB

(6) BC and

(7) AC

The file 'INPUT' will have the following entries.

Table A.2

Full adder 'INPUT' file for the Stick Diagram Generator

Line No.	Contents
1	3 7 2 5.0 0.0 2.0
2	ABC (Input variables list)
3	ABC (1st prod. term)
4	AB'C'
5	A'BC'
6	A'B'C
7	AB
8	BC
9	AC
10	1 2 3 4
11	5 6 7

Total No. of lines = No. of prod. terms+No. of outputs + 2.

= 7 + 2 + 2 = 11.

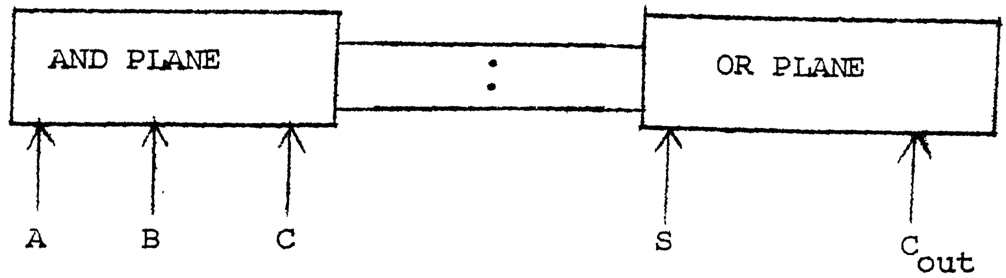


Fig. A.2 Block schematic of the stick diagram for the Full adder

After creating the INPUT file and copying the files STK, MAT.PAS and STICK.MIC, the stick diagram is generated by the following monitor command

.DO STICK.

A.3 PLA Layout Generation

All the subroutines needed for the generation of layout is in the file named 'LEOUT' (Note that it is not LAYOUT for reasons which become evident later on). The PASCAL program 'MAT.PAS' producing the personality matrix and the file 'LEOUT' are linked in the file 'LAYOUT.MIC'.

The INPUT file for the PLA layout is identical to that of the INPUT file for the Stick-diagram Generator.

After creating the INPUT file and copying the files 'LEOUT', 'MAT.PAS', and 'LAYOUT.MIC', the PLA layout of the

desired logic function is obtained by the following monitor command,

.DO LAYOUT.

For more examples on INPUT file creation refer to the Chapter 5 - 'Results and Conclusions'.

APPENDIX B

PLOT-10

The PLOT-10 Interactive Graphics Library (IGL) is a host-independent library of routines for graphic and text interaction. Programs calling IGL routines may be written in any language that can call a FORTRAN library subroutine. The IGL source code is written in IGL MORTRAN and is compiled into FORTRAN when IGL is installed.

The following routines have been used in the implementation of the packages discussed in this thesis.

GRSTRT (4010,1)

This must be the first call in any block, which makes calls to Interactive Graphics Library (IGL) routines. The numbers (4010,1) refer to the graphics display device.

GRSTOP

This must be the last subroutine call in any block having calls to IGL subroutines.

MOVE (X,Y)

The cursor moves to the point P(X,Y).

DRAW (X,Y)

Draws a line from the current cursor position to the point P(X,Y).

POLY(N,A,B)

Draws a polygon of N sides. The X co-ordinates of the corner points of the polygon must be in the array A and the Y co-ordinates in the array B.

DASHPT (I)

Draws a line (continuous, dotted, alternating dash and dot, etc.) according to the value of the integer variable I ($0 \leq I \leq 9$).

ROTATE (X_{rot}, Y_{rot})

Rotates the X and Y co-ordinates by the corresponding angles X_{rot} and Y_{rot} respectively.

TRANSL (X,Y)

Translates the current origin to the origin P(X,Y).

SCALE (X_{scale}, Y_{scale})

Scales the X and the Y axes by the factors X_{scale} and Y_{scale}.

VECREL

Treats the subsequent displacements in the X and Y directions as relative displacements with respect to the current cursor position.

WINDOW (X_{min}, X_{max}, Y_{min}, Y_{max})

Defines a window, the lower left corner at (X_{min}, Y_{min}) and upper right at (X_{max}, Y_{max}).

VWPORT (X_{min}, X_{max}, Y_{min}, Y_{max})

Defines a viewport with lower left corner at (X_{min}, Y_{min}) and upper right at (X_{max}, Y_{max}).

MILLIM

Defines that all values are in milli-meters.

INCHES

Defines that all values are in inches.

RASTER

Defines that all values are in 'Raster' units.

GDUNIT

Defines that all values are in Graphical Display Units.

RADIAN

Defines that the subsequent angle specifications are in Radians.

DEGREE

Defines that the subsequent angle specifications are in degrees (default).

PIVOT (X,Y)

Defines that the point about which scaling and rotation are about the point P(X,Y).

EDGE (X_1,X_2, Y_1,Y_2)

Defines the clipping window with lower left corner at (X_1,Y_1) and upper right at (X_2,Y_2).

CLIP

Defines the clipping window as the viewport.

NOCLIP

Defines that no clipping is to take place at the viewport.

FILPAN (IP,Q)

IP = Pattern number

Q = .TRUE.

= .FALSE.

Defines the pattern (such as horizontal hatching, vertical hatching, etc.) to be used to fill a polygon. If Q = .TRUE. the boundary of the polygon is outlined and will not be outlined if Q = .FALSE.

PANEL (N, X, Y)

Fills the polygon with the pattern defined by the FILPAN subroutine. N is the number of sides of the polygon and X and Y are arrays containing the x and y co-ordinates of the corner points.

SKIP

This usually precedes the call to the poly subroutine. This causes the cursor to move from the current cursor position to the first corner point of the polygon.

For more IGL routines reference may be made to the PLOT-10 IGL Manual.

APPENDIX C

SPICE - The Circuit Simulator

SPICE is a general purpose circuit simulation program for nonlinear dc, nonlinear transient, and linear ac analyses. Circuits may contain resistors, capacitors, inductors, mutual inductors, independent voltage and current sources, four types of dependent sources, transmission lines and the four most common semiconductor devices : diodes, BJT's JFET's, and MOSFET's.

Circuit description : The circuit to be analysed is described by a set of element cards, which describe the circuit topology and element values, and a set of control cards, which define the model parameters and the run controls.

Example : The following deck determines the transient characteristics of a NMOS inverter with depletion load, feeding a load capacitance of 5 pF.

```
C1    2   0   15PF
M1    3   2    2   0           MDP L = 20U    W=5U
M2    2   1    0   0           MEN L = 5U     W=5U
VDD   3   0   DC        5V
VIN  1    0   PULSE  OV  5V   15NS   15NS   15NS   3US   6US
.MODEL   MDp   NMOS   (KP = 25    E-6,  VTO = -4V)
.MODEL   MEN   NMOS   (KP = 30 E-6,  VTO = 1V)
.TRAN    15NS   1.5US
.PLOT   TRAN    V(2)
.END
```

The first 3 lines of the above deck give the nodes between which the elements are connected, i.e. C1 between nodes 2 and 0, VDD supply of 5 Volts between nodes 3 and 0, MOS transistor M1 between nodes 3,2,2,0 (Drain, gate, source and substrate respectively), and M2 between 2,1,0,0. Vin is a pulse of OV to 5 Volts with a delay time of 15nsec, rise time of 15nsec, fall time of 15nsec, pulse width of 3 micro-sec., and period of 6 micro-sec. The first MODEL statement indicates that the circuit element is an NMOS transistor with a process gain factor of 25 x 10^{-6} and threshold voltage of -4V. The second MODEL statement refers to the enhancement MOS transistor (driver). The next statement states that transient response from zero - 1.5 micro-sec,in steps of 15nsec is sought for. The last but one card defines the node at which the transient response is needed.

SPICE input decks of many other circuits can be found in the 4th Chapter (on simulation).

For more details reference can be made to the SPICE manual.